MICRO AUTONOMOUS SYSTEMS RESEARCH:

SYSTEMS ENGINEERING PROCESSES FOR MICRO-AUTONOMOUS SYSTEMS

GEORGIA INSTITUTE OF TECHNOLOGY

DR. K. DANIEL COOKSEY
DAVID LOCASCIO
ERIC SCHAUS
ZACH FISHER
ARTHUR CHENG
RAPHAEL GAUTIER
LUQMAN OLAWIN
PETER MANGUM
NOHTAL PARTANSKY
DR. DIMITRI MAVRIS (PI)

SEPTEMBER 30, 2015

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
30-09-2015	Annual Research Report	20141001 - 20150930

4. TITLE AND SUBTITLE
Micro Autonomous Systems Research
Systems Engineering Processes for Micro-Autonomous Systems

5a. CONTRACT NUMBER
LaRC/ARL Coop NNL09AA0A

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
K. Daniel Cooksey, David Locascio, Eric Schaus, Zach Fisher, Arthur Cheng, Raphael Gautier, Luqman Olawin, Pete Mangum Jr., Nohtal Partansky, Dimitri Mavris

5d. PROJECT NUMBER
2A31

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Aerospace Systems Design Laboratory, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
RDRL-VTV
6340 Rodman Rd
Aberdeen Proving Ground, MD 21005

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This material is based on work supported by the National Aeronautics and Space Administration, Langley Research Center under the Research Cooperative Agreement No. NNL09AA00A awarded to the National Institute of Aerospace. Any opinions, findings,

14. ABSTRACT
In Phase II of the Micro-Autonomous Systems Research project, the Georgia Tech teams developed an automated product family engineering process and toolset allowing the creation of tailored one-off solutions to soldier needs. The toolset provides a simplified user interface for non-technical users to enter vehicle requirements, such as sensor packages, endurance, payload, etc. A spreadsheet logistics interface allows an untrained logistics operator to enter machines and parts availability. This information is fed to set of engineering analyses where a feasible design (if possible) is generated, and the drawings for manufacture are output. These part designs are then provided to a technician with automated manufacturing tools (such as 3D printing) who starts the automated manufacturing, assembles components, and returns the tailored UAV to the soldier. This process has been tested and validated via flight tested vehicles and satisfies the desire to be more responsive to soldier needs for small unmanned aerial systems.

15. SUBJECT TERMS
unmanned systems, design space exploration, systems engineering, integrated design and manufacturing, additive manufacturing, 3D printing, on-demand, forward deployed, conceptual design, sizing, architecture, experimentation, scenario simulation, technology tradeoffs, fixed wing, multicopter, quad copter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Eric Spero
Unclassified	Unclassified	Unclassified	UU	71	19b. TELEPHONE NUMBER *(Include area code)* (410) 278-8743

Reset

CONTENTS

INTRODUCTION

In Phase II of the Micro-Autonomous Systems Research project, the Georgia Tech teams developed an automated product family engineering process and toolset allowing the creation of tailored one-off solutions to soldier needs. The toolset provides a simplified user interface for non-technical users to enter vehicle requirements, such as sensor packages, endurance, payload, etc. A spreadsheet logistics interface allows an untrained logistics operator to enter machines and parts availability. This information is fed to set of engineering analyses where a feasible design (if possible) is generated, and the drawings for manufacture are output. These part designs are then provided to a technician with automated manufacturing tools (such as 3D printing) who starts the automated manufacturing, assembles components, and returns the tailored UAV to the soldier. This process has been tested and validated via flight tested vehicles and satisfies the desire to be more responsive to soldier needs for small unmanned aerial systems.

PROJECT HISTORY

A vital requirement of the modern combat environment is to gain and maintain situational awareness to facilitate effective squad-level decision making. Over previous years, Georgia Institute of Technology (Georgia Tech) has collaborated with the Army Research Laboratory (ARL) in developing design capabilities to assess the operational capability of micro autonomous vehicles to assist at the squad level. Improved systems engineering processes for micro-autonomous systems is the primary focus of the research undertaken in the Micro-Autonomous Systems Research MASR effort. This report details the work done in Phase II of the MASR effort undertaken as a joint effort between ARL and Georgia Tech.

Phase I is focused on the development of the systems engineering processes necessary for the development and test of an autonomous system for use within a building's interior. The emphasis of Phase I was on the creation of a vehicle via a rigorous systems engineering process and led to the creation of a flying prototype capable of mapping the interior of a room.

Phase II began upon completion of phase I, and the emphasis shifted towards the acceleration of the previous systems engineering process for rapid deployment in response to changing soldiers' needs. The systems engineering process developed in Phase II, was achieved through the use of extensive modularization of the micro autonomous systems, which was combined with automated modeling, design, and manufacturing tools.

PHASE II MOTIVATION

The overall objective of the ARL-Georgia Tech collaboration is to improve squad level situational awareness and effectiveness via small autonomous systems. However, the requirements for squad level needs span a broad range of scales relative to the size of the vehicle. For example, the soldier may desire a miniaturized camera that can be mounted on a very small vehicle that can be piloted through a building. Alternatively the soldier may desire a sensor package such as the one used in Phase I that can map that building. In terms of requirements, the payload of a camera would only way a few grams, the endurance may only need to be a few minutes, and the size will be limited by the doors and windows of the building to be searched (less than 20 inches and potentially smaller). However, the soldier may also desire a vehicle capable of providing continuous moving surveillance (scouting for a convoy) for an hour or more, but have no size limitations. The same soldier on a differing mission may want to fly a medical kit from an outpost to a team on a patrol, and may need to carry a few pounds for a number of miles, with the ability to precisely deliver the payload. One of the technological challenges at the squad level is that mission needs can evolve on a day-to-day basis, and is complicated that the mission needs may be unforeseen at deployment. At the micro-autonomous system scale these requirement changes have significant impact on the asset required.

Three generalized approaches can be employed when trying to develop an asset that best satisfies the diverse set of soldier needs.

- **Multi-mission asset:** You can generate one UAS which is able to cover all mission needs but at the cost of sacrificing performance on some missions.
- **Set of optimized assets:** A set of optimized vehicles can be used which are able to perform several missions very well, but here you run the risk of troops needing to carry an overwhelming number of assets to cover all mission possibilities
- **Asset On-Demand:** this "on-demand" philosophy would grant the soldier access to a very wide range of UAV assets which are flexible enough to adapt to unforeseen needs

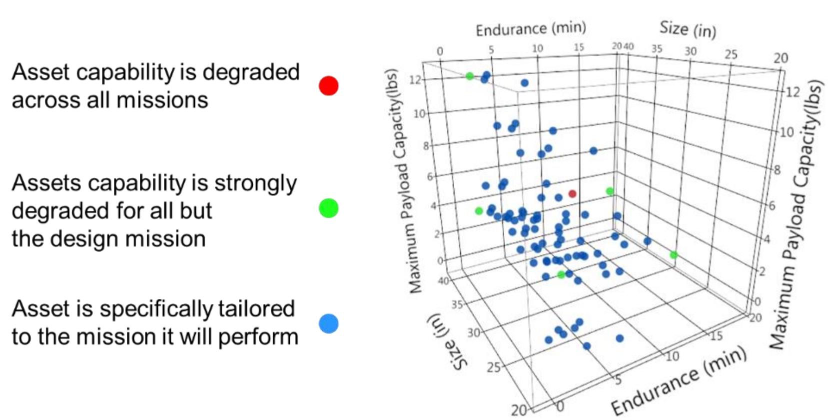


Figure 1: Three approaches to satisfying soldier needs

Figure 1 shows three example dimensions defining a soldiers requirements. The three differing generalized approaches to addressing the broad set of requirements are plotted in the differing colors. The red dot shows a multi-mission asset centered in the middle of the space. While the green dots show a larger coverage by making four or five tailored assets available. However, this approach implies a larger logistical burden on the unit. The final generalized approach implies that the soldier can build or adapt to the need as it arises. The blue dots in Figure 1, represent this "on-demand" approach. This approach implies some level of access to rapid manufacturing and rapid engineering. Traditionally, technical hurdles have limited an "on-demand" approach.

However, the advent of new manufacturing technologies, such as additive manufacturing, offer the potential for combat invention, innovation, modification, and manufacture to be forward deployed enabling an "on-demand" asset. This new capability enables improved soldier flexibility to create materiel solutions to the problems they face. The historical process for deploying materiel solutions requires the soldiers waiting for materiel solution requests to propagate back up the chain of command, be relayed to the industrial base, which must be spooled into production, with the solution eventually deployed back to the warfighter. However, the new manufacturing techniques, such, as 3D printing, consumer-focused automated CNC milling and laser cutting enable simplified construction of one-off systems and parts tailored to the situation at hand. These new manufacturing techniques have the potential to enable the soldier to rapidly produce the answers to their problems as a stop gap while the more traditional manufacturing processes are getting to speed.

Figure 2: Forward Deployed Manufacturing Lab (http://www.ref.army.mil/exlab.html)

This vision has the potential to drastically improve the space of solutions available to the soldier and improve army agility and combat capability, but a number of research challenges remain in realizing this vision. It has been recognized that the rapid manufacturing toolset currently being developed for commercial purposes could be adapted for military use, and can be forward deployed. Moving the tooling forward toward the soldier is currently being implemented with the Army's Rapid Equipping Force (REF). The trained technicians in these forward deployed fabrication facilities have the skillset to use the tooling they have been provided; however, it is not typical for the average soldier to have the engineering analysis expertise or time necessary to develop an advanced unmanned system in response to a mission need. The Phase II of the ASDL-ARL collaboration has been to research the applicability of an engineering process and toolset capable of enabling the soldier to move from mission needs to an engineered UAS solution.

OBJECTIVE

The phase II objective included developing an engineering process that provides the soldier a way to enter their needs and then return to the soldier an Unmanned Arial Vehicle (UAV) tailored to those needs. This objective was decomposed into the more actionable objectives of providing: 1) a simplified interface for entering UAV requirements, manufacturing resources, and parts availability which will drive 2) the developed engineering process and 3) returns to a trained technician the files for manufacture.

This top level objective centered on the user must be supported by a back-end analysis. The back-end includes a number of automated analyses for systems, mechanical, and electrical engineering. These analysis have been combined with a specified set of forward deployed sub-systems, and their accompanying 3D models, that can be integrated with 3D printed parts and assembled similarly to Lego® bricks. This enables the rapid design and manufacturing of small UAVs for improving battlespace awareness in a capability referred to as "In-situ Design on Demand".

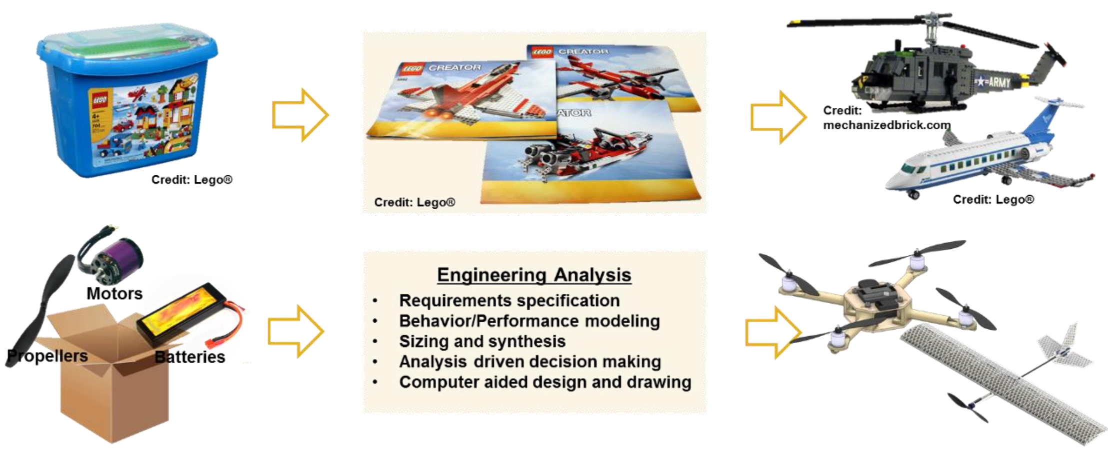


Figure 3: Lego Analogy for Defining Objective

Figure 3 succinctly summarizes the objective of the Phase II research using an analogy to Lego® bricks. Lego® bricks contain a number of modular parts that can be constructed into differing models depending on what is desired. Instructions are provided to help the user build differing systems. In an engineering context, a small set of parts will be provided to the REF which cannot be manufactured on hand. These parts along with additional ones that can be manufactured will be combined to create the needed system. The objective for this research is to provide the analysis and instructions (in the form of engineering files) that allow the technician to take the parts and assemble the tailored UAS.

MEASURES OF SUCCESS

The team's success has been measured by comparing the output process to the requirements of each of the stakeholders in the chain necessary to go from soldier requirements to a delivered UAV.

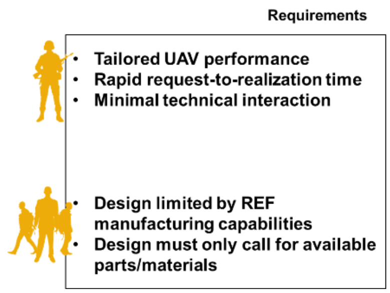


Figure 4: Stakeholder Requirements

Figure 4 shows the two sets of stakeholders in getting a feasible product to the soldier. From the soldier's perspective, the UAV must meet the requirements requested (or inform the user that no current design can meet those requirements). The design must be produced in a rapid time frame (a matter of days), and provide information on how long manufacture will take. Finally, these two elements must be captured in a format that requires little technical interaction from the untrained soldier. From the REF technician standpoint, the design must match the manufacturing capabilities on hand, and it should only use materials and supplies that are available. The process developed was measured in its ability to meet these different requirements and the outcomes are discussed in the results section.

PHASE II: APPROACH

OVERVIEW

The approach developed for meeting the described requirements has been termed "**A**ggregate **D**erivative **A**pproach to **P**roduc**t Design**" (ADAPt Design). In this approach, a product design is synthesized through the combination of a selection of a set of off-the-shelf or "modular" components, and the integration of those components via a set of scalable, adaptable components. The result is a UAV design that is made up of a subset of the parts in the deployed in the kit of parts brought together by parts manufactured for the specific use. The net result obtained is this custom tailored UAV design.

MATRIX OF ALTERNATIVES AND MISSION NEED COVERAGE

The stated goal is to provide a tailored UAV to cover the mission needs of the soldier. To accomplish this feat, it is necessary to create the broadest set of alternatives possible. In Phase I of the MASR project, the team used the concept of an Interactive Reconfigurable Matrix of Alternatives to help enumerate the space of possible solutions. A matrix of alternatives was developed by the team to help enumerate the possible space of combinations for a limited set of components that could fit in a slightly larger than shoe box sized container.

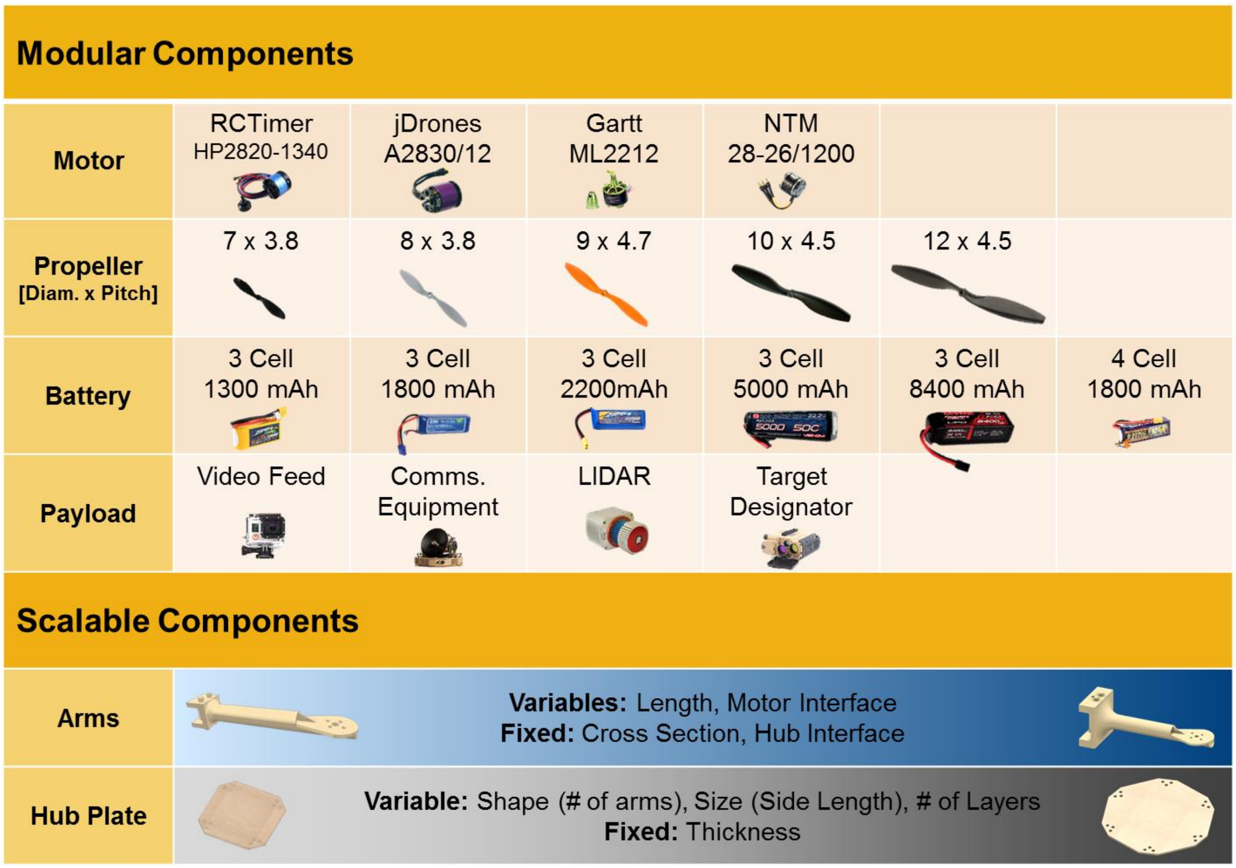


Figure 5: Matrix of Alternatives for Micro Autonomous Systems

The limited set of components shown in Figure 5 can combine to create 480 discrete alternatives sets of items, which can be combined with the continuous components to provide a broad coverage of the mission space.

PRODUCT FAMILY DESIGN

In development of the ADAPt Design approach, the team employed techniques borrowed from the design of product families. Much of the product family literature focuses on single platform product families, where the product are made to be either modular or scalable. For the purposes of this report we will define a platform as "a set of components and subsystems that form a structure from which a number of product variants are derived"[1]. From this platform, modular product families focus on swapping individual modules to create derivate designs. Alternatively, scalable platforms focus on scaling elements of the design to create derivatives. To capture as broad a set of soldier needs as possible, the ADAPt Design approach will use both product family strategies in the creation of product derivatives.

Beyond the creation of a set of derivatives for a single platform, the team created a multi-platform product family where two platforms, a multirotor and a fixed wing, share a set of common parts from which each platform and it's derivatives can be derived. Each of these platforms is also a hybrid platform containing both modular and scalable pieces. This is motivated by the requirement for tailored performance and aims for max flexibility by varying multiple aspects of the platform.

DESIGN AUTOMATION

While elements of the product family approach can help in the development of a family of designs to cover a broad range of soldier needs, satisfaction of the speed and simplicity measures of success identified requires that the product family design process be automated.

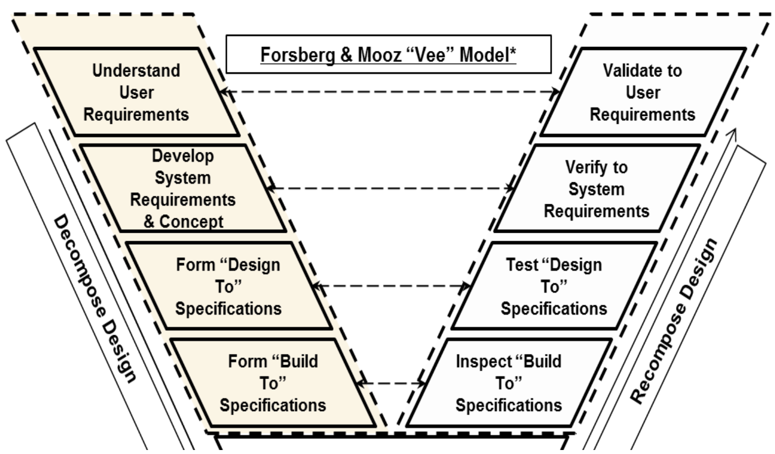


Figure 6: Systems Engineering Vee Model

To set up the ADAPt Design approach, it can be useful to compare it to a traditional development process: the systems engineering "vee" diagram.

Starting on the left leg of Figure 6, the vee model first breaks down the mission needs and requirements through analysis into sets more and more detailed specifications to which the individual parts of the UAV can ultimately be built. The first difference in the traditional approach and our approach is based on the need for the soldier to be removed from technical activities. This means that the left side of the vee, which contain the engineering activities, will need to be automated.

The right side of the vee involves activities that validate the items designed and built meet the specifications and the original mission need. In normal development, this leg usually requires the bulk of the time for the entire process, and can be very slow due to the pace of testing and redesign. Because the process is limited to a few days by our requirements, the activities in this side of the vee will need to be bypassed. That is – the design produced must have to be very high likelihood that it will meet the soldier's needs without any verification or validation. However, factoring in the requirements of the logistics chain at the REF, one finds this problem is unique in that the following two items are known ahead of time: what the automated design process is capable of producing and which components will be available for use. These two items have been leveraged to largely pre-validate the right leg of the vee during design automation by incorporating validated modeling and experimental results. Through the following elements 1) the high precision of the Automated Manufacturing processes used, 2) Virtual Assembly of the UAV, 3) prior analysis of the components in the kit, and 4) simulation, the error that creeps into the engineering process can be limited to a point where validation can be bypassed.

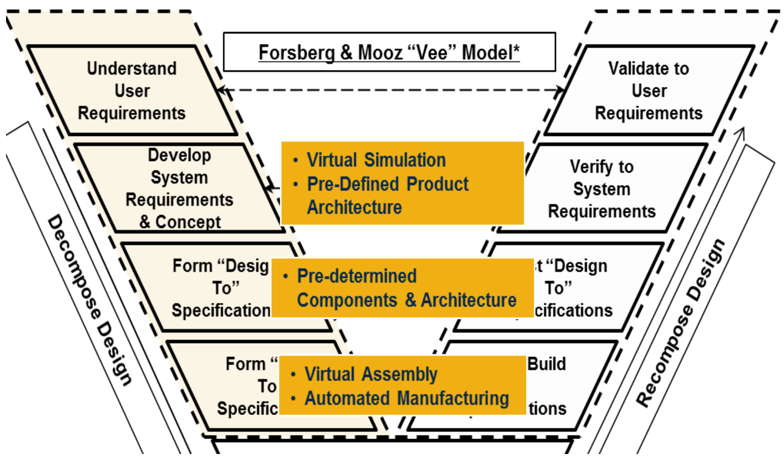


Figure 7: ADAPt Elements Collapsing the vee Model

Figure 7 shows the systems engineering vee models with a number of elements which enable the ADAPt Design process placed on top of the model. The ADAPt Design process uses validated and verified engineering models to link the right and left half of the vee collapsing the vee horizontally. These validated models have been automated to the point that they can move directly from requirement to Computer Aided Design (CAD) files that can be directly ingested by the automated manufacturing tools allowing the vee to be collapsed vertically. The details of this are described in the next section.

ADAPT DESIGN PROCESS

The previous sections have focused on building the background and philosophy of the creation of the ADAPt process. The following sections will detail a process for implementing the ADAPt Design process.

The core of the ADAPt design effort revolves around using rigorous systems engineering techniques to link requirements to design intent to the output design in an executable manner. It is important to note that in the previous sentence the word executable is not intended as the currently fashionable programmatic buzzword. In the context of this work, executable indicates that the logic linking these elements has been documented in executable code. This code used to directly drive Computer Aided Engineering and Design (CAE\CAD) tools which can directly output drawings for manufacture.

For simplicity's sake, the ADAPt design process which leads to modular design has been documented in a series of linear steps. However, like all design processes, the acquisition of information in the ADAPt Design process occurs over time and previous decisions should be revisited in an iterative manner.

The following sections will use the development of a modular set of multirotor vehicles and fixed wing aircraft to illustrate the design process. The process was developed around the multirotor and tested against the fixed wing aircraft design using a different toolset. The current section focuses on the process but details of the vehicle test can be found in the "Testing and Results" section.

STEP 0: REQUIREMENTS ANALYSIS

The zeroth step in the ADAPt design process is to gather and understand the range of needs that the modular product family will address. These should then be linked to engineering metrics, objectives, and constraints. These metrics objectives and constraints will eventually inform the automatic selection of alternatives by the design environment.

The approached recommended by the Georgia Tech – ARL team is to define the capability that is to be achieved. This includes articulating clearly whose needs will be addressed and some statement of how they will be addressed. Achieving any new capability often requires support of multiple stakeholders when logistical, political, technological aspects are considered. The second step is to identify stakeholder's needs, requirements, and constraints and use these to bound the product family design. Success can be defined in this context, as finding a set of solutions that simultaneously satisfies all of the stakeholder's needs.

EXAMPLE VIA CASE STUDY

As stated above, the capability desired was the ability to supply the soldier with a tailored UAS solution to an unforeseen need. Much of this zeroth step was documented in the introduction section, but more details are included here for completeness.

For the second phase of the research effort, knowledge about potential needs from Phase I of the MASR effort was used to inform the requirements generation. As part of Phase I, five missions where Micro-Autonomous Systems could provide small unit support were identified.

- Convoy Surveillance & Defense
- Perimeter Surveillance & Defense
- Building Interior Reconnaissance
- Cave Interior Reconnaissance
- Jungle Reconnaissance

From these missions, a set of fixed payloads could be identified, and they have been enumerated on Figure 5 on the payloads row. Furthermore, range of other performance metrics, (range, payload weight, endurances, etc.) could be identified that could be paired with the payloads to meet the needs for the identified missions. As described in the motivation section, the performance requirements, shown graphically in Figure 8 drive this particular problem towards an on-demand capability.

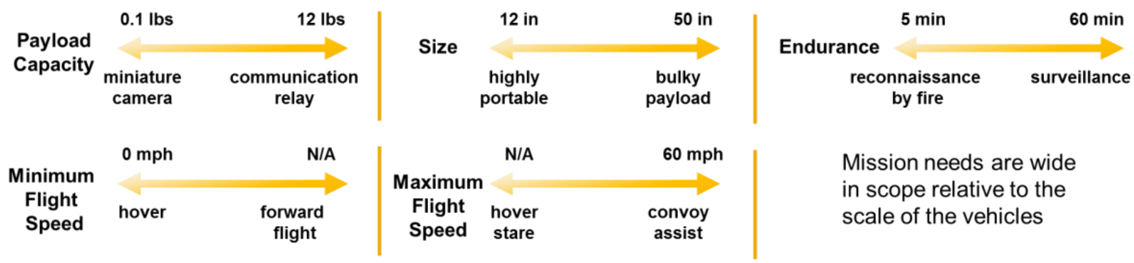


Figure 8: Vehicle Performance Requirements

The second step was to examine any follow on effect that may occur via stakeholders required to implement the capability described. The on-demand capability requires additional manufacturing and logistical considerations. In this case, the Rapid Equipping Force, was identified as an existing group within the US Army that could satisfy those considerations, but introduced constraints of their own. The designs produced must fit in the tooling they have, and must conform to the set of items they have on hand. These requirements were used as objective and constraints in bounding the architectures selected.

STEP 1: ARCHITECTURE SELECTION

The goal of the architecture selection is to identify the platform or platforms on which the modular family of designs will be built. This has been broken into sub steps listed below. For each step in the ADAPt Design documentation below, the activities required are described first. At the end of each step a section dedicated to describing these steps in the context of an implementation is provided. This breakdown allows this documentation to easily be used as both an informative and reference article.

1.1 PLAFORM IDENTIFICATION

The first step in the architecture selection is platform identification, which will identify the platforms from which the derivative deigns will be constructed. Three methods are recommended for identifying promising platforms.

1. List functions required to complete the mission, match components to these functions, and then build vehicle alternatives from these components via a morphological matrix
2. Identify classes of vehicles that historically have been used to cover parts of the desired mission space, and determine which regions have no current coverage.
3. Brainstorm single vehicle concepts that address extreme edges of the requirements space with an emphasis on the regions without current coverage.

The team recommends using the results of each of these three techniques in a manner that allows ideas from one to be used to inform and update the others. The output from this exercise is a set of concept platforms that meet the requirements.

1.2 FUNCATIONAL DECOMPOSITION

Once the concept platforms have been identified, the functional roles of each of those platforms should be enumerated. Most of this work will have been completed in the first brainstorming step 1.1, but revisiting the functions should be done with the platforms in mind can help identify functions that have been missed. All of the functions required for the platform to fulfil the mission requirements should be identified.

Once the functions have been identified, matching components and sub-systems to the functions can accelerate the development of a modular multi-product family. Each function should have some sub-system or component that fulfills that function. Particular emphasis should be placed on identifying existing off-the-shelf components and sub-systems that can fulfil these functional roles.

1.3 COMMON FUNCTIONS AND COMPONENTS

The goal of step 1.3 is to identify locations where common components may be used across the different platforms. The first step is to identify common functions across the platforms used to fill the requirements space. Once this has been done, the platforms components and sub-systems should be examined to determine if they can fulfill jointly shared functions. It is also important to identify the set of components and sub-system which are unique to each platform at this stage.

1.4 BUILD COMPONENT LIBRARY

At this point, the designer should set up the component library. This is a library of components or sub-systems, which will contain key information on how the components are to be modified (modular or scalable), the interfaces for each of the components, each component's functional role, along with other engineering data that may be useful in deriving new designs. The data elements listed in the previous sentence and placed in the component library will be populated throughout the ADAPt Design process. The component library is used to track how differing components are being used and modified, as well as to provide a resource for other projects building from the same components. The designers also recommend allowing for data fields specific to the fulfillment of a particular function be entered into the component library.

1.5 IDENTIFY COPONENTS AS SCALABLE OR MODULAR

For each of the components identified in step 1.3, determine if the component will be modular or scalable. Modular components are those in which discrete alternatives are swapped. Scalable components are those in which selected dimensions of a component are varied across a continuous rage. An example of modular components would be the use of differing off-the-shelf motors. An example of a continuous component, would be a multirotor arm which could be changed to match the design needs.

The last element that is important to note, is that the decision for a parts modular or scalable designation can be nested. For example, the selection of a wing sub-system might be considered modular because the selection of the airfoil is discrete. However, within this selection the span of that wing may be scalable. While the selection of the airfoil and the span influence each other, one decision must be made first, and as a result, it is recommended that the component library reflect this decision logic rather than listing the part as a hybrid component.

1.4 BACK OF THE ENVELOPE BASELINE PLAFORM DESIGN

At this point in the process a "back of the envelope" platform design should be conducted (similar to traditional conceptual design), if it has not already been done when trying to identify promising candidates for fulfilling the requirements. This back of the envelope design should be based on engineering intuition and contain some analysis that helps inform some of the basic bounds that will be passed to the detailed design of the interfaces.

The initial concept design study should not conform to the traditional idea that "at the end of conceptual design, broad conceptual changes to the design are frozen". The purpose of the ADAPt Design process is to create a family of concepts based around common components. Instead, this initial concept design should provide the designer the following: 1) an idea of which parts requirements will drive the interfaces 2) how broad of a range of changes the interface will experience 3) a decent idea of where all of the interfaces and components will be places.

1.5 CREATE INITIAL PLATFORM SKELETON

Using the information from the back of the envelope platform design and the divisions from the component and function matching, an initial skeleton model should be generated. The skeleton model is a model where the key geometric planes, points and shapes are located in space. This skeleton will pass these key planes, points and shapes to the individual parts. This skeleton modeling process when integrated with a CAD program has been branded "top down design" by some in the community.

The use of this skeleton has two uses: the first is that it helps the designer define the boundaries of components, and the second is that it helps to remove iterative cycles from the 3D modeling part of the design process. Updating the 3D model to converge on a set of components that fit together can be very slow, but the skeleton can be iterated on quickly. This model will be heavily refined, but explicitly stating the key divisions being used helps to organize the process.

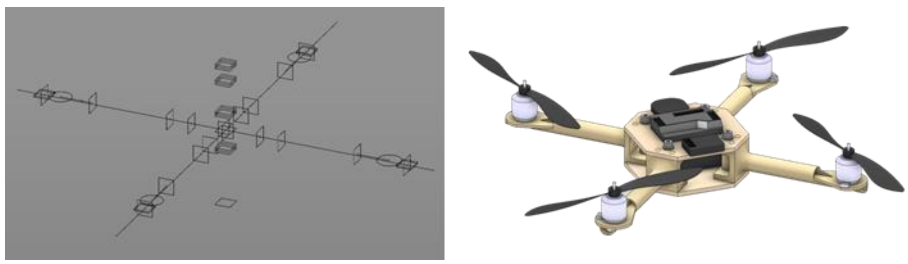


Figure 9: Skeleton Model and the Vehicle it Represents

The skeleton model generated at this point doesn't necessilarly have to contain all of the key points or elements that are likely to appear in the final and could even be simply sketched on paper. However, the identification of which elements will be tracked as the underlying structure of your platform, and how those elments relate to eachother should be identified at this point.

EXAMPLE VIA CASE STUDY

The requirements for the design of a modular "on-demand" family of UAVs to support the soldier were listed in the phase 0 section. To meet these requirements, the team had created a morphological matrix of the functions necessary to meet these requirements and matched these to a broad set of subsystems in the MASR Morphological Matrix of Alternatives delivered in Phase I of this research. The team also examined the existing platforms for small unit ISR, and from this analysis, the team identified two promising candidates for platform architectures.

- A multicopter design to cover the hovering applications
- A hand or ground launched fixed wing design to cover the longer endurance applications

Figure 10 shows the functions and the components for the two platforms identified. In the center of Figure 10, the components have been grouped in the by role. In the yellow box, are the components shared by both platforms. In the white bottom center box, the components that are unique to the platforms are listed. At this point in the process, the team categorized all of the shared components as modular, because they were all off-the-shelf, and all of the unique components as scalable. These designations are also shown in the center of the figure. However, at later iterations of the design process the wing would be made into modular component because the first decision made was discrete (airfoil), and then the discrete alternative would be scaled from there.

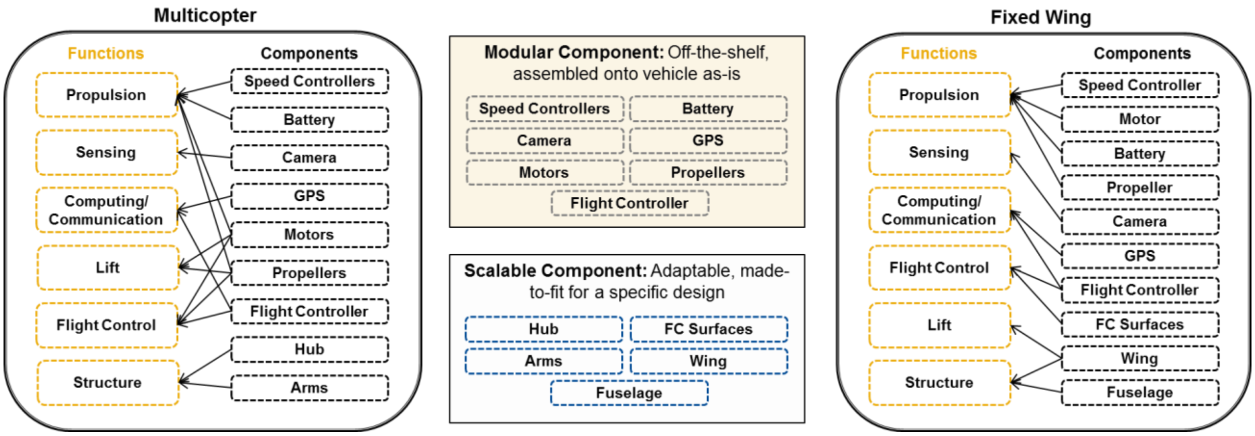


Figure 10: Functions and Components for the Multicopter and Fixed Wing Platforms

The team built a component library in an excel spreadsheet, which was determined sufficient for tracking. In more complex problems, a database could also be used.

Figure 11 shows the kind of information stored in the component library. At this point in the process, the component library entries for the off-the-shelf components were fully entered into the library. The parts that were to be created for the specific designs were given a place holder, and filled into the component library later. However, at this point each component had an entry and had a preliminary designation as modular or scalable. The team also later found it useful to link this excel sheet to the location where we had stored test data for each of the differing prop and motor combinations.

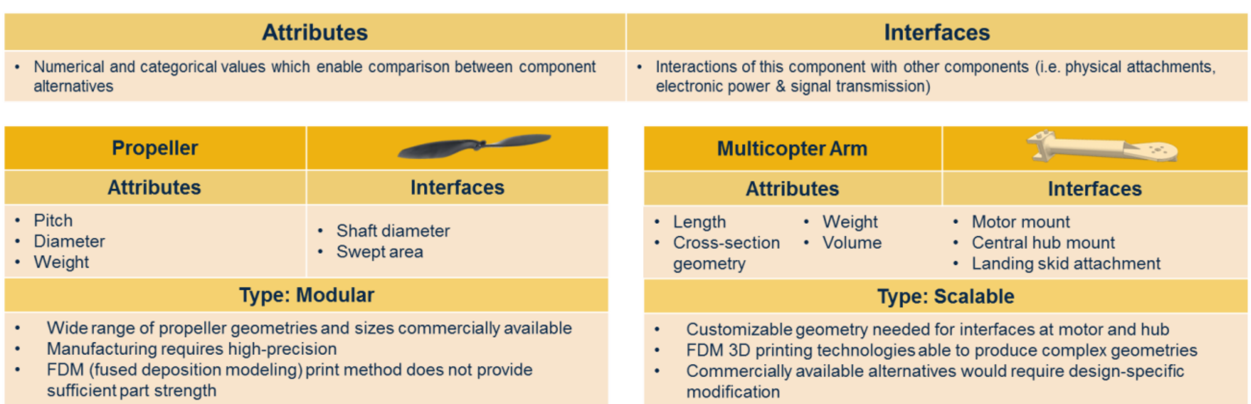


Figure 11: Component Library Information

Next, the team performed a zeroth order sizing analysis of the multi-copter and the fixed wing aircraft. These analyses were used to determine the maximum and minimum sizes of certain components and interfaces. For example, this analysis gave us an initial estimation of how large a prop and motor would be required on the larger multicopter vehicles. This in turn let to information about the requirements for strength on the interface between the arm and the hub. Figure 12 shows an example of some early multicopter concept design work. From this figure, it can be seen that the idea of using plates with attached arms had been decided, but the interfaces and arm designs had not been settled.

Figure 12: Early Concept Design Work

Finally, a skeleton for each of the models were developed. These skeletons had key information about where one modular part would interact with another. For the multicopter, the skeleton was created in CATIA V6 as an assembly file, which could be updated by an external sizing program and allowed changes to be propagated to parts automatically. For the fixed wing aircraft, parametric parts were modified within a set of CREO Parametric V3 CAD files but the key locations for the skeleton were contained in a separate Matlab file. Both methods were successful, but the skeleton integrated with the 3D model files allowed for more convenient refinement of the design, and there seemed to be real value in executing the design process using what most CAD software vendors refer to as "top down design". It is important to note, that the first skeleton was nearly completely rebuilt as iterations of the design were completed, so the designer should not be striving for the perfect division between components at this point.

STEP 2: INTERFACE DESIGN

One of the key and novel differences between the traditional engineering design process and the proposed process built for modularity is the early emphasis on interfaces. Through this research effort the team determined that prior conducting full scale conceptual design, the designer should conduct detailed design on the key interfaces, enabling modularity.

The iterative nature of design processes in general, meant that at times the detailed interfaces needed to be updated, but the early emphasis on the interfaces provided a standardized mechanism by which new candidate designs could be algorithmically generated.

STEP 2.1 STANDARDS IDENTIFICATION

The first step in the interface design is to determine the interfaces for the off-the-shelf components. These interfaces are by definition fixed and any standards in these interfaces should be identified and added to the component library.

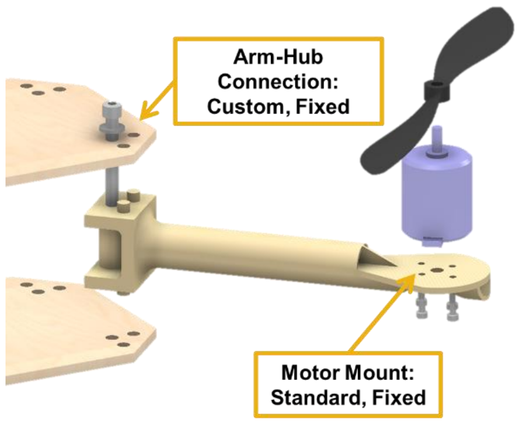


Figure 13: Example Interfaces

Figure 13 shows an example of two mechanical interfaces. One is a standard defined by the electric motor being used and the other is a custom interface that has been defined as standard by the team. At this point in the off-the-shelf standard interfaces should be added to each component in the component library.

STEP 2.2 CUSTOM INTERFACE CREATION AND STANDARDIZATION

The second element to the interface design is the creation of custom standards for interfaces. The use of the skeleton and the zero order platform analysis should have provided some idea about the range of options the interface will need to accommodate. Using this information a standardized interface should be created which will be applied to all of the future evolutions of a particular platform.

However, unlike the standardized interfaces from the off-the-shelf components, the custom interfaces can contain additional degrees of freedom. As part of the interface, rules (captured in a computer code) can be defined that specify how an interface will change as the parts around it are changed. These rules can be tracked in the component library, or some other modeling tool. For the example discussed below, the rules (along with the geometry) of the interface were contained in a CAD model, linked to code that would update the interface in response to changes in the skeleton.

STEP 2.3 SKELETON MODEL UPDATE

A further definition of the interfaces leads to a new set of points or planes that may need to be incorporated into the skeleton model. These points and planes are at the boundary of two separate components or sub-systems, and as a result likely need to be managed centrally from the skeleton. As a result, it is necessary to revisit the skeleton once the interfaces have been designed.

STEP 2.4 INTERFACE TESTING

The final step in the detailed design of the interfaces, is to test the interfaces at the extreme ranges of the changes they are expected to incorporate. For standardized interfaces for off-the-shelf components, this can be as simple as virtually swapping the components and making sure that the interface automatically updated on the other parts to match. For scalable, and custom defined interfaces, the use of the first pass platform modeling of is used to define the test ranges for the interface.

EXAMPLE VIA CASE STUDY

The following paragraphs will discuss the way in which the interface definition and design was completed for the MASR multicopter architecture. Figure 14 shows three of the components for the multicopter and their respective interfaces.

The first two components shown in Figure 14 were standard off-the-shelf components, and each belonged to a class of similar components. For example, the modeling environment contained 4 possible motors of differing types, and 5 propellers of differing types. Each of the motors each had a shaft diameter, bolt patter, current draw for differing voltages, and wire plugs as interfaces.

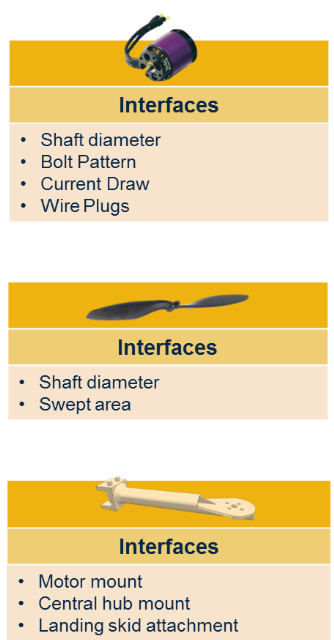


Figure 14: Three Components with Their Respective Interfaces

The interfaces listed contain not only the physical interfaces, but also electrical, and interference interfaces. For example, the motor had two standard physical interfaces for the bolt pattern: a small pattern and a large pattern depending on the size of the motor. The motors also had a maximum current draw which represents one of the electrical interfaces. This interface changed for each motor and was checked automatically later by the vehicle sizing to ensure an appropriate motor selection. Referring to the second component in Figure 14, the propeller, there is an interface listed for swept area. This was the circular area that the blades would rotate through, and was maintained as an interference interface so that the 3D model could verify that no other components were within this area and would be impacted by the propeller blades as they spin.

The third component in the list shown in Figure 14 is the multicopter arm. This arm contains two interfaces: the hub connection, and the motor connection. A clear depiction of these interfaces can be seen in Figure 13. The motor interface was created so that the bolt hole pattern on the arm could be automatically updated to match that of the motor, once one was selected. The interface at the root of the arm where it attached to the hub plate as shown in Figure 13 was under full control of the designer. This single interface contains many dimensions and variables, and to ensure that the design process can be automated and solved via computer in a reasonable time, many of the degrees of freedom needed to be fixed via a standard. The team eventually decided on an interface for the root of the arm that fit into three holes that would be place on the hub plates. One hole would have a bolt through it and the other two would have locking tabs which extended from the arm. This pattern became the standard interface between a single hub plate and the arm. However, during the initial concept design, it had been determined that the battery should be placed between these two plates. As a result, the size of the battery (plus a defined margin) was set to define the height of the arm root interface and each arm would scale to match the vehicle needs. All of this was documented internal to CATIA V6 within the KnowledgeWare programming language (the CAD program being used for this design). To do this the team created a parameter that controlled the height of the root for the arm. This height was then set to be driven by the distance between two planes (which represented the hub planes) in the skeleton model.

Next, the skeleton model was revisited to ensure that parameters, such as the distance between the hub planes, existed in the skeleton and could drive the rest of the design.

Finally, the interfaces were tested to determine if they would respond appropriately to changes in the design across the ranges that were expected from the initial concept modeling. For example, the root interface had to be tested to ensure that a very small battery or a very large battery would not break the rule set defining the vertical interface, and that a small hub plate for a very size constrained multicopter would not push the hole pattern to the point where the holes overlapped.

STEP 3: CONCEPT REFINEMENT AND DESIGN

The concept refinement and design step is where many of the elements thought of as traditional engineering design occur. However, rather than using a set of engineering analysis and tools to support design decisions, in the ADAPt Design process these analysis and tools will be linked together. This allows the design decision making to be automated so that the design process can operate in an automated fashion.

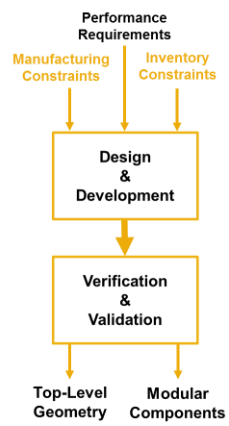


Figure 15: Simplified Design Process

Figure 15 shows a highly simplified design process. The top box represents the engineering analysis and the second half represents the test of that analysis. The next sections will address these two elements of the design process.

However, the basic two activities shown are supplemented by a series of elements which are used to enable the "on-demand" vision. Typically, the only input to the development process is a set of performance requirements. The manufacturing setup will be tailored to the design that is to be produced, and the inventory will be purchased to match the design being produced in a typical design process. However, in the "on-demand" model, every time a new design is generated these two constraints must be considered, as the most desirable manufacturing and inventory my not be available in this process. In addition to the design logic typically considered in conceptual and preliminary design, manufacturing and availability constraints must be incorporated at these early phases of design. The final element required beyond traditional design is that the designer must drive the sources of error in the engineering models to extremely low levels. This reduction of error in the models used for design is especially important in the ADAPt Design process, because the new vehicle automatically created by the design environment will not pass through any verification and validation phase. As a result, the models used in automatically developing a design must have very low error. Essentially the burden of verification and validation is passed from the assembled subsystem or vehicle, to the engineering models so that they can be reused in an automated fashion.

STEP 3.1 CONSTRAINT IDENTIFICATION

One of the first steps performed in the development of the design is an identification of design constraints. For this exercise, the team should examine the constraints on the design typical of a normal engineering process. Most of which were addressed in the initial sizing exercise. The team should also perform a brainstorming activity to determine constraints to the design that could occur via the logistical chain, which would lead to a number of rules and considerations about how less than optimum parts can be mixed when needed. Next the team should perform a brainstorming activity to identify the constraints that would occur due to changing (or less than optimum) manufacturing equipment and how the design should be constrained or adapt in these cases. Finally, the team should conduct a brainstorming activity to determine how differing parts would impact other parts. As an example, often times the determination of the location of bolt holes is left until late in the design process, and as a result, the consideration for tooling (wrenches, Allen keys, etc.) that can reach the bolt head during manufacture is something that is only considered as the final touches are being made on the detailed design. In an automated design process, these considerations must be taken into account as the design is being developed. These considerations have to be documented in a series of checks embedded in the automated design code. The team should attempt to identify these at an early stage and build a check list of these considerations. Furthermore, these brainstorming activity can be conducted in a relatively short amount of time, and the team found it useful to repeat it for each new sub-system or part being completed. The four categories used in brainstorming are shown in Table 1.

Table 1: Brainstorming Categories

Constraint Brainstorming Categories	Example
Design Constraints	Range, Payload, etc.
Logistics Constraints	Part Availabilities, Assembly Expertise, etc.
Manufacturing Constraints	Machine Bed Sizes, Bit Availability, etc.
Part to Part & Assembly Constraints	Wire Routing, Assembly Interactions, etc.

STEP 3.2 MODEL BASED DESIGN & DEVELOPMENT

Figure 16 shows a generalized modeling process for the ADAPt Design process. For clarity's sake, this section has been broken into the following two elements: 1) conceptual and preliminary modeling, which is modeling necessary for the updating of the skeleton and top level parameters of the design, and 2) the detailed design modeling, which brings a part to the point of manufacture.

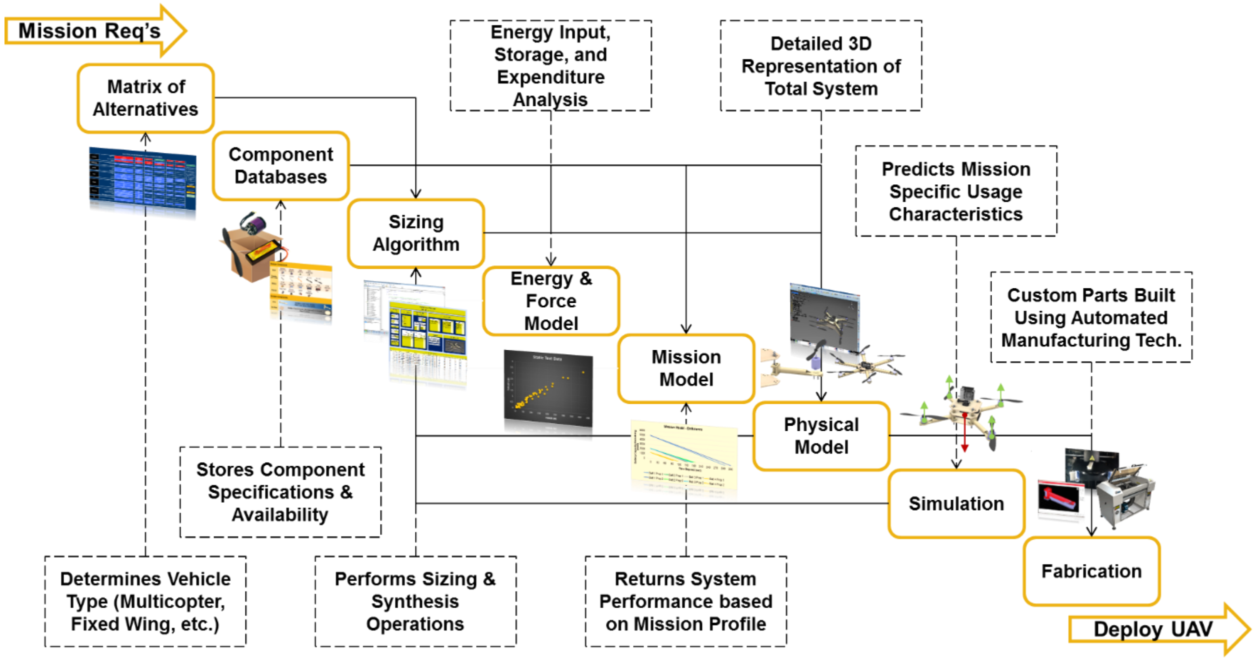


Figure 16: Generalized Modeling Process

The steps moving from the requirements to the mission model are considered the "conceptual and preliminary modeling environment". One 3D modeling of the physical components is taking place, this is considered "detailed design modeling".

This chapter will not go into the specifics of the design and modeling as these should be problem dependent. However, it should be noted that the process has been tested on two separate UAV designs, and is believed to be generalizable.

STEP 3.2.1 CONCEPTUAL AND PRELIMINARY MODELING

Returning to Figure 16, a generalized modeling flowchart for the ADAPt Design process on can observe that the modeling process takes in the mission requirements, uses a Matrix of Alternatives to determine the vehicle class, and then proceeds to progress through a fairly standard modeling process. The modeling process will refer to the component library. The first step in the modeling process is to perform a physics based sizing algorithm. This could be the same one used in the quick look sizing described in step 1, or it could be a higher fidelity sizing algorithm. The second step is to balance energy and force. For the case of the UAV's presented as part of this work, the energy and force balanced included developing a model of the propeller, its forces, power draw, and a model of the power system and its capacity. The next piece of the integrated modeling environment was a mission model where the vehicle could be tested to ensure it met all the critical mission constraints throughout the mission. These elements conclude the conceptual and preliminary modeling.

This conceptual and preliminary modeling should provide the high level inputs to be fed forward to the detailed modeling through the use of the skeleton model. Since the purpose of conceptual and preliminary design in the typical engineering environment is to select and propagate top level engineering dimensions along with part level requirements to the detail design process, the conceptual and preliminary modeling step has been modified in only two specific ways. The first is that all of the tools must be linked in software to ensure they run in a single integrated environment. The output of this integrated model must also be automatically passed to the detailed design modeling and this has been conducted through the use of a skeleton model. The second element that differs is that these models also must be verified and validated to ensure that the error is very low. This second element is discussed in Step 3.3.

STEP 3.2.2 DETAILED DESIGN MODELING

The detailed design modeling piece creates parametric 3D models of each individual part backed by a ruleset that can adjust those parameters. To accomplish this the conceptual modeling environment should be linked so that it can automatically update the skeleton model. Furthermore, the interfaces identified in Step 2 of the process should be placed within the CAD. A series of logical rules and design iterations will also be performed to optimize and ensure that the final geometry meets the constraints and requirements. Figure 17 shows an overview of this process.

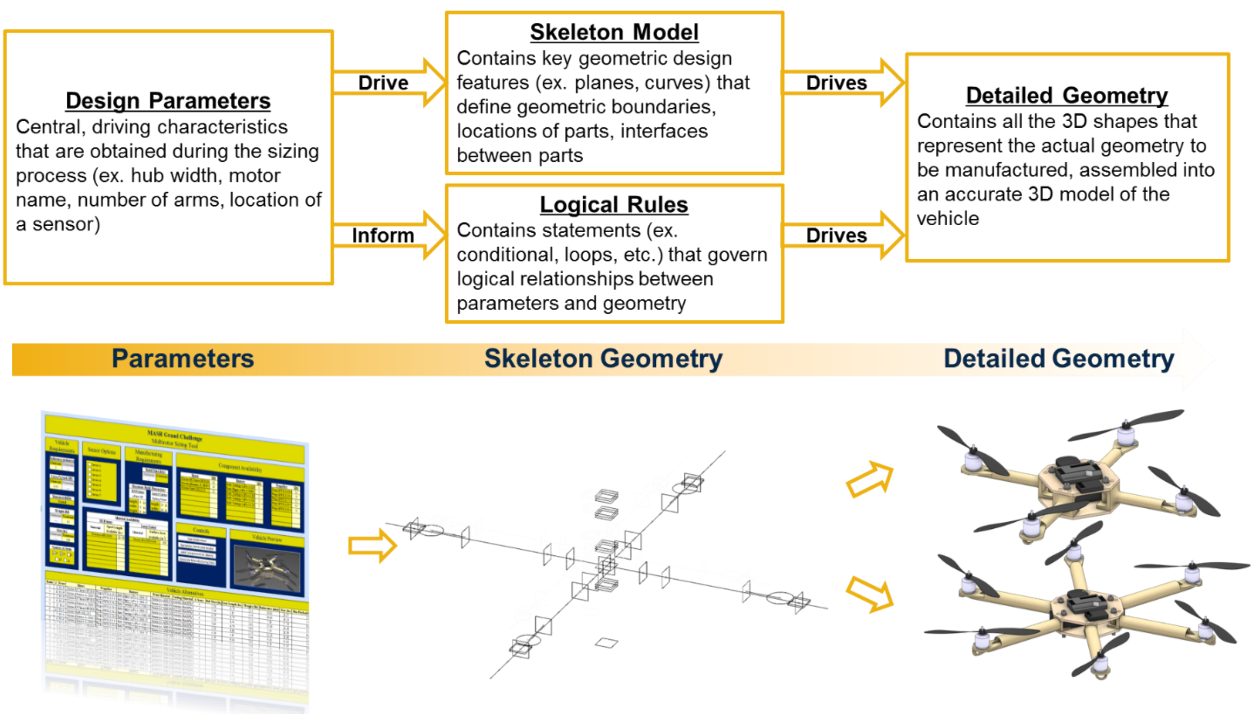


Figure 17: Process for Driving Geometry via Modeling Environment

The second step in detailed design modeling is to update the detailed geometry. For the initial iteration, the detail geometry of each part must be created. To create the geometry of each part, the modeling process describe above was repeated at a lower level. The skeleton and conceptual models set the requirements for this lower level design iterations. For example, the skeleton will drive elements such as the length and strength required of specific parts. This then starts a lower level modeling process which begins with the brainstorming of constraints and progresses through updated geometry. The updated geometry can be a lower level skeleton or the actual part. The process is repeated moving lower and lower into the design until automated process is driving specific dimensions and parameters of a part for manufacture.

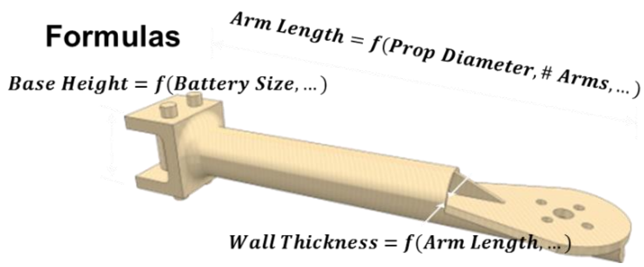


Figure 18: Multirotor Arm Example

Figure 18 shows an example of a single arm for the multicopter. This part is representative of the process. For this part, the base height (the vertical height of the block at the root of the arm) was driven by the skeleton via the hub planes which were derived from the battery size. Within the CAD model for this part, this base height dimension was left as a parameter, which could be changed and the geometry could be updated. In this case, the update propagates from the battery selection to the skeleton, through to this part and modifies the geometry. The same process is used for the arm length which is set to be the shortest length where the propellers have clearance from each other and the center hub. This arm length, along with the force from the motor will determine the moment that the arm has to carry, and as a result, the wall thickness will also need to be updated. By leaving all of these elements as parameters, the part can be automatically updated as the vehicle is required to increase its payload.

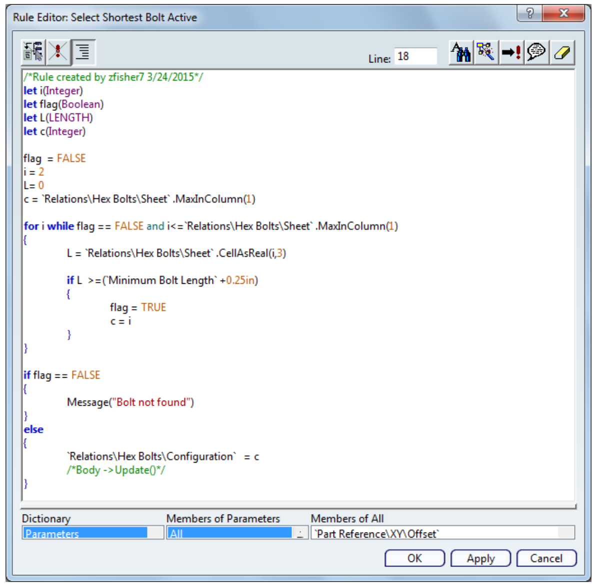


Figure 19: Example Program for Model Update

Figure 19 shows an example of how these rules have been coded into CATIA V6 KnowledgeWare. In this case the rule shown is looking for a bolt that can fit the hub plate and arm assembly. If a bolt is found it will update the model to contain the correct bolt. If not, it will display a message for the user stating that no bolt is found.

As long as the design process for the differing parts is conducted in a way that these parameters are left exposed for modification, the vehicle design can be updated to match new requirements. With the individual parts completed, the individual parts can be virtually reassembled, the 3D modeling environment can be used to pull global variables that can be difficult to calculate individually, such as center of mass. Furthermore, assembled design can be tested in a simulation of the desired mission. Ideally this would incorporate the actual flight software in a 3D simulation environment. Once the design has been virtually validated, the design should be passed to the automated manufacturing technicians. This design should be output from the detailed modeling environment in the format that the automated manufacturing toolset ingests.

STEP 3.3 MODEL VALIDATION

Throughout the modeling and design process, it is vital that the models for the vehicles and individual parts be validated for the range on which they will operate. For the purposes of the ADAPt Design process, the model must be validated across the whole expected range of the design space. Typically this can be accomplished by testing the extremes and a few of the center points. This high degree of validation of the modeling environment is necessary because the design will not be tested before operational use.

The use of standardized modules and interfaces however, provides a mechanism by which the validation of the models can be broken into individual elements. If each component is valid within the boundaries of its interfaces to a high degree of confidence, then it is likely that the assembled model will perform as expected in operation.

EXAMPLE VIA CASE STUDY

The first step in the modeling process was to brainstorm constraints. The team performed this exercise and identified a number of constraints with respect to how the layout should be structured for both the fixed wing aircraft design and the multicopter design. As an example, one of the key constraints for both architectures was the 3D print bed tray size. In both cases, this size limited the maximum dimension of any part, leading to a multi-section wing on the fixed wing aircraft. On the multicopter it this constraints limited the length of the arms and the multicopter hub needed to be allowed to scale up when a larger clearances between propellers was necessary.

The second step within the concept refinement and design was the creation of the conceptual and preliminary modeling. Figure 16 shows a generalized modeling process which begins with a morphological matrix. Figure 20 shows the interactive morphological matrix created for this design problem. The matrix shown was created as part of Phase I of the MASR effort. Using this tool along with the items described in Step 1 of the ADAPt process, the team had determined that the most appropriate architectures for exploring further development would be a hand launched fixed wing aircraft and a multicopter UAV. For the case study developed in Phase II, the morphological matrix was not directly linked via code to the modeling environment, but it could have been used as a method for selecting architectures or discrete components.

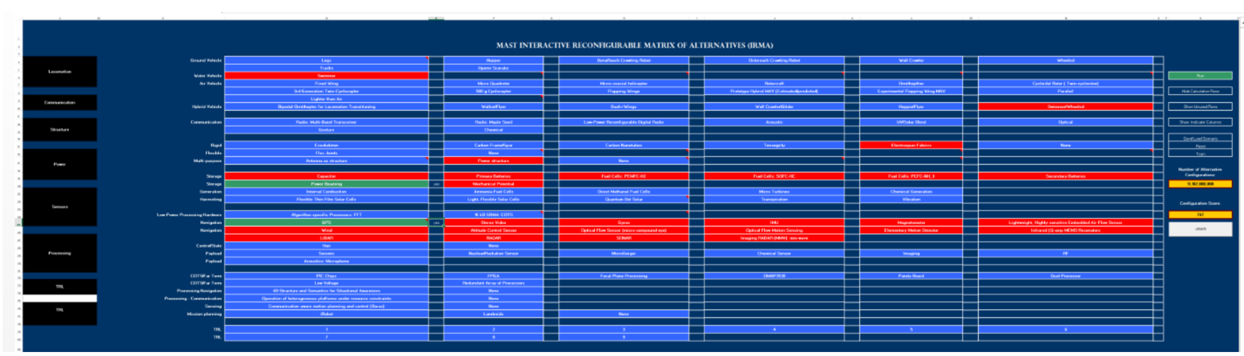


Figure 20: Morphological Matrix

After selecting the classes of vehicles for further study, the team returned to the component library and constructed it in a manner that it could be linked to the rest of the modeling environment. Each of the CAD packages surveyed could ingest a table from Microsoft Excel. As a result, the component library was built in Microsoft Excel. Figure 21 shows the battery data for the component library. It contained information on the type of battery, along with critical information such as the weight and dimensions that were used in automatically generating a CAD model to match the specifications. The component library for the battery also contained interface information such as capacity, maximum allowable current draw, and the voltage. Similar information was stored for other off-the-shelf components such as motors and propellers.

Reference	Manufacturer	Model	Battery Type	Weight (lbs)	Cell Count	Capacity (mAh)	C Rating	Dim x (in)	Dim y (in)	Dim z (in)	Return to Dashl
Batt-Turnigy-LiPo-3-2200-1.5	Turnigy	9XR	LiPo	0.2875	3	2200	1.5	4	1.25	0.625	As
Batt-Zippy-LiPo-3-5000-25	Zippy	lightmax	LiPo	0.8375	3	5000	25	5.5	1.875	0.875	
Batt-Turnigy-LiPo-4-1800-50	Turnigy	unlabeled	LiPo	0.4875	4	1800	50	4.125	1.25	1.125	
Batt-Turnigy-LiPo-3-2200-50	Turnigy	unlabeled	LiPo	0.4375	3	2200	50	4.125	1.25	1	
Bias-Single	Bias	lightmax	LiPo	1.18125	4	5000	50	5.25	1.8	1.8	
Batt-Turnigy-LiPo-3-8400-50	Turnigy	nanotech	LiPo	1.41	3	8400	50	6.18	1.77	1.77	
Bias-Parallel	Bias	lightmax	LiPo	1.18125	4	10000	50	5.25	3.6	1.8	
Bias-Stack	Bias	lightmax	LiPo	1.18125	4	10000	50	5.25	1.8	3.6	

Figure 21: Component Library Battery Data

Figure 22 shows an image of the thrust data from the component library for each of the propeller and motor combinations. It is important to note that this type of engineering information was also stored within the component library where possible. The sizing algorithms used in developing the multirotor pulled this data in developing the design.

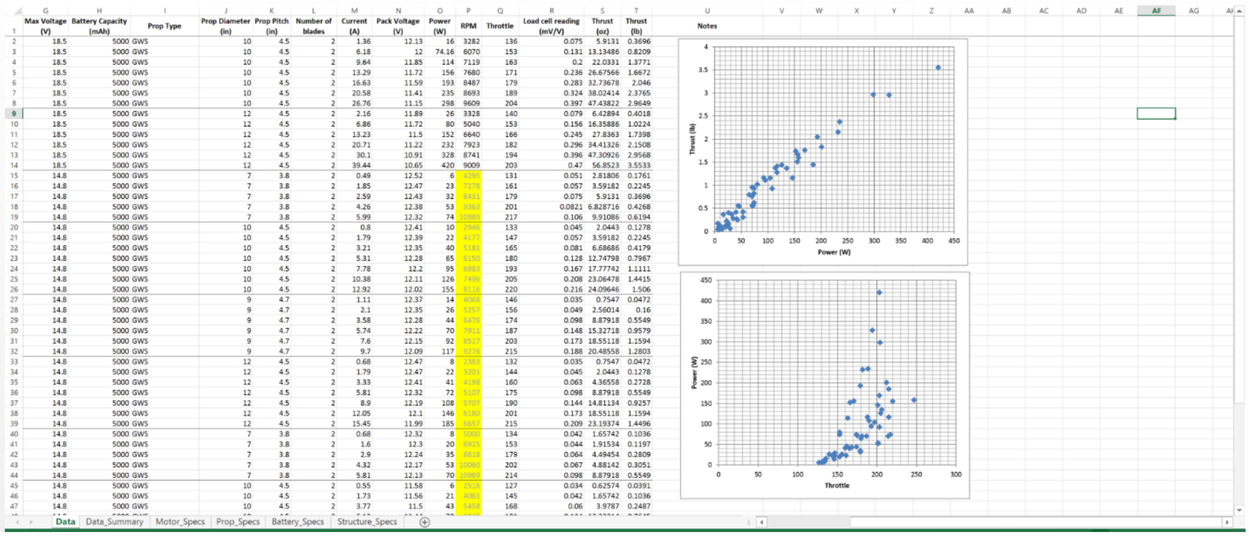


Figure 22: Component Library Thrust Data

Figure 23 shows a screen capture of the sizing algorithm for the multicopter vehicle. The team developed a sizing tool in Microsoft Visual Basic to more easily integrate with the component library. The team developed a separate sizing tool for the fixed wing aircraft within Matlab. This tool also linked with the external programs Xfoil and AVL for aerodynamic analysis during the sizing process. Each of these methods pulled components from the component library. Having the component library directly integrated into the modeling, as well as keeping it separate from the modeling were implemented with relative ease, and whichever best suites the designers needs is recommended.

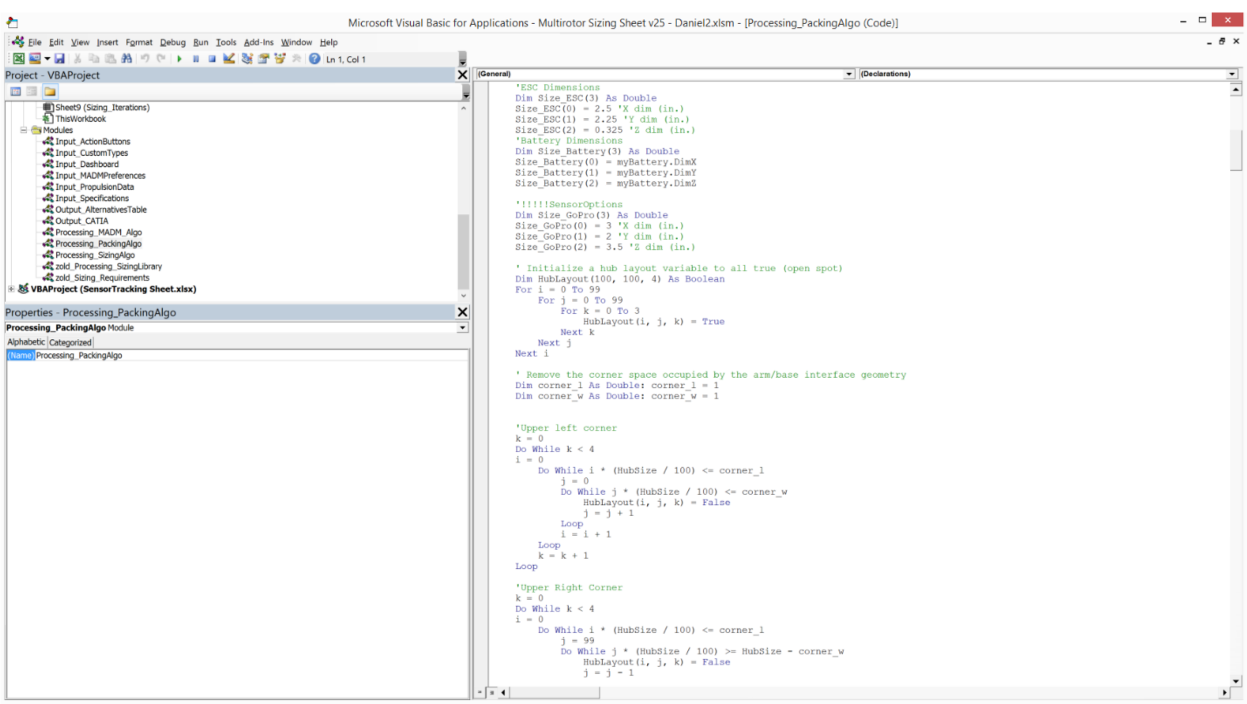


Figure 23: Multirotor Sizing Algorithm

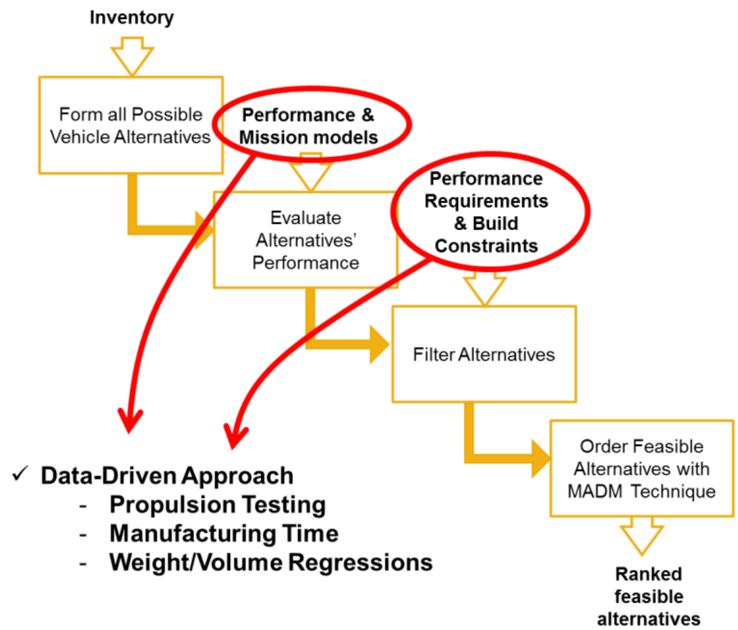


Figure 24: Multirotor Vehicle Selection

The algorithm shown in Figure 24 is the routine that is responsible for selecting the combination of modular parts to update the multicopter skeleton, which drives the parameters needed to scale scalable parts. This algorithm generates all possible feasible combinations of multi-copter and then orders them by goodness.

The structure of the algorithm was mainly driven by two challenges specifically:

- The algorithm must handle the need to simultaneously operate on modular and scalable parts. The algorithm forms all the possible designs based on all combinations of modular parts, and then the total designs are filtered based on the mission requirements and estimates of each designs performance. This setup addresses the issue of differing types of parts by letting the algorithm operate by always matching scalable parameters to a selection of modular parts.
- The removal of the Verification & Validation phase implies that the models must be particularly accurate. A data-driven modeling approach was chosen because physical phenomena based models that are usually used for rotorcrafts, such as blade element theory, did not yield high enough accuracy results at this small scale. The experimental validation of the models used is discussed later.

The process described allowed the design automation framework to generate all possible designs for the multi-copter that could satisfy the mission requirements. The set of designs capable of meeting the requirements ranged from there being no possible solution to having dozens. In the case where multiple feasible solutions were returned the multicopters were ranked using the Technique for Ordered Preference by Similarity to Ideal Solution (TOPSIS). This tool requires information on how different attributes are preferred relatively to each other, and the TOPSIS algorithm was made emphasize a satisfaction of requirements with a maximum endurance time. All solutions capable of meeting the requirements were returned to the user as ordered by TOPSIS so that the user could select the design the user preferred.

The design of the fixed wing vehicles, example shown in Figure 25, proceeded in the opposite manner than the multicopter. For the fixed wing aircraft, the key dimensions (wing area, span, etc.) were not constrained by off the shelf components. As a result, the continuous dimensions were determined first, and then modified to match a motor and battery. Discussion of this process in described in the results section. Both the approach of allowing off the shelf components to constrain the tailor made components and allowing the tailored made components to select the closest off the shelf components were successfully implemented within the modeling process.

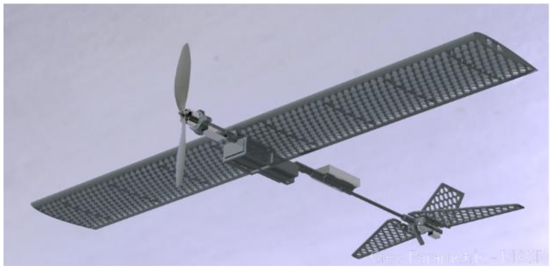


Figure 25: Member of the Fixed Wing Class of Vehicles

Figure 24 refers to a data driven modeling approach. This reference is intended to highlight that the modeling for the multicopter was originally performed using physical theory based analysis such as Blade Element Momentum Theory (BEMT) and Finite Element Analysis (FEA). Initial tests to validate these models found they were inconsistent with the results for this scale of vehicle and selection of materials.

Figure 26 shows an image of the thrust test stand used to generate the data captured in the component library and shown in Figure 22. Because the BEMT analysis had performed poorly, the model for thrust, power consumption, and other propulsive analysis were replaced by a regression model of the measured thrust data.

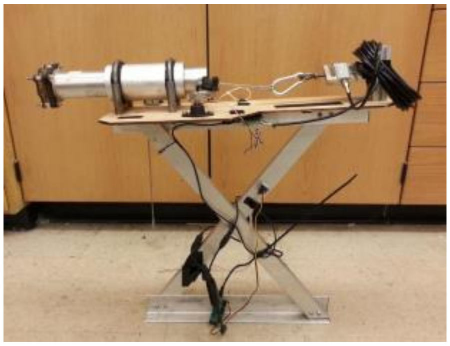


Figure 26: Thrust Test Stand

It was also observed that the FEA models traditionally used for modeling structures performed poorly for the 3D printed materiel. The non-uniform Fused Deposition Modeling (FDM) materiel deformed and failed significantly sooner than the FEA modeling suggested. Figure 27 shows an example of the structural testing conducted on representative multicopter arms.

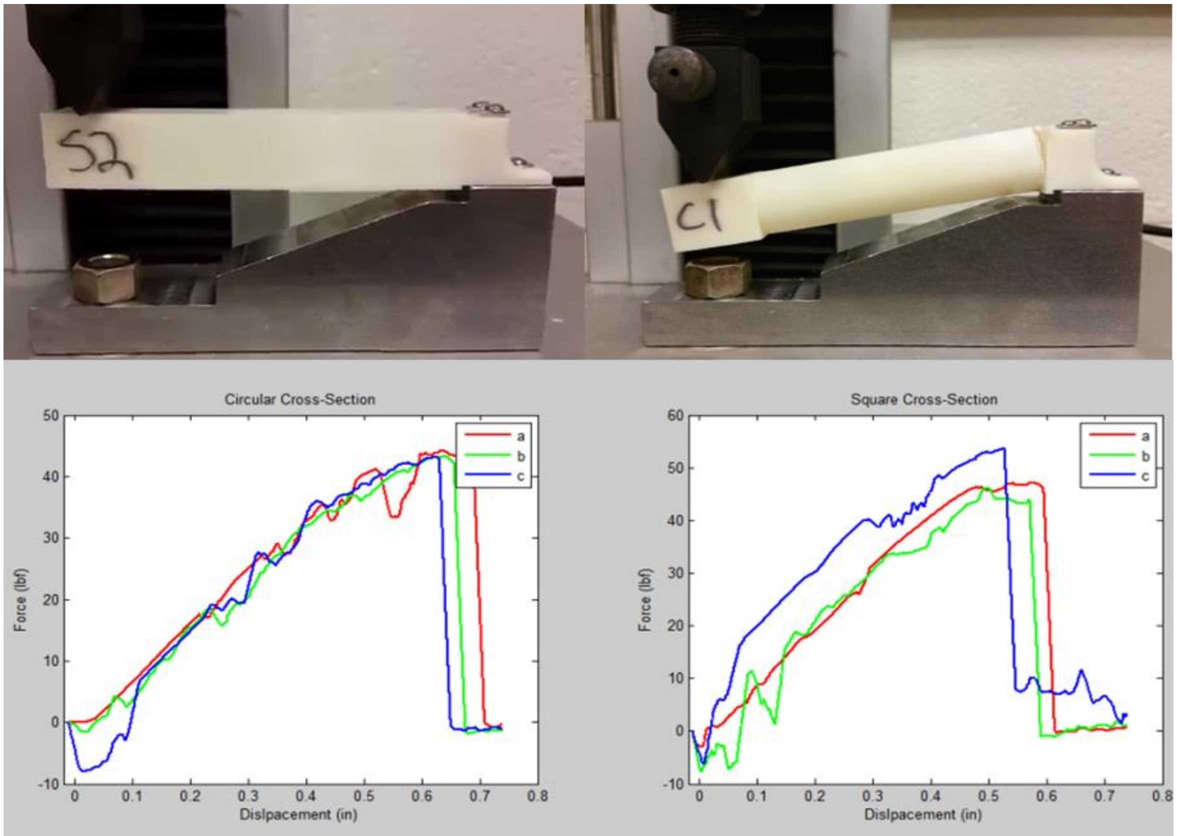


Figure 27: 3D Printed Beam Load Test

Figure 28 shows a comparison of the FEA results and the actual load tests. Figure 29 shows the comparison between the FEA results for the circular cross section and the experimental results. The experimental results were regressed to determine a model for use in the modeling environment.

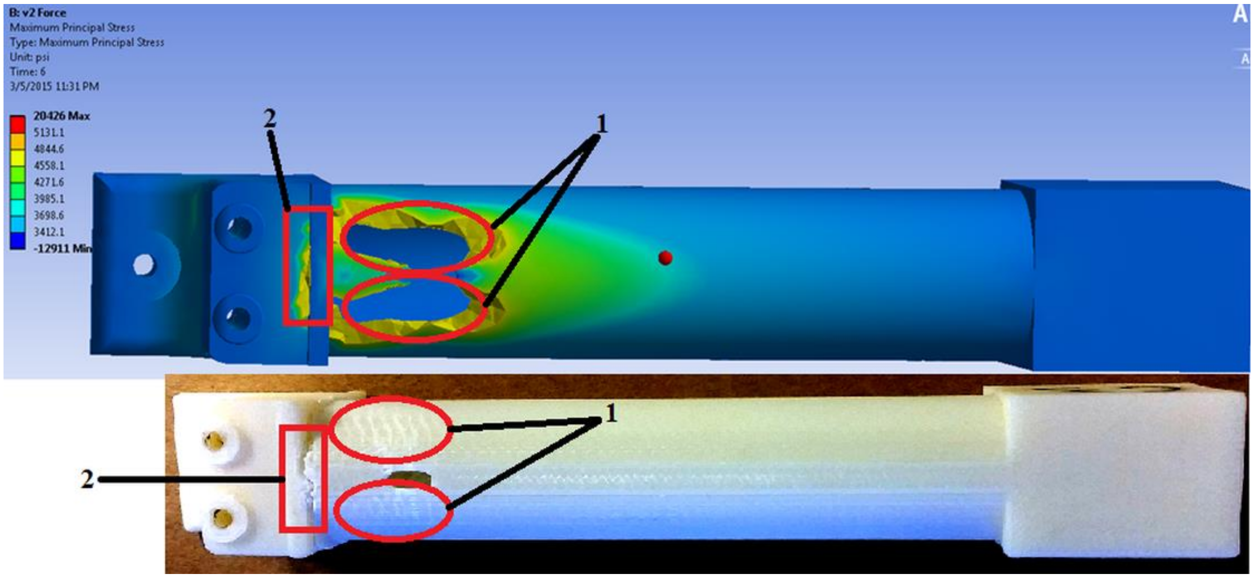


Figure 28: Comparison between FEA and Test Article

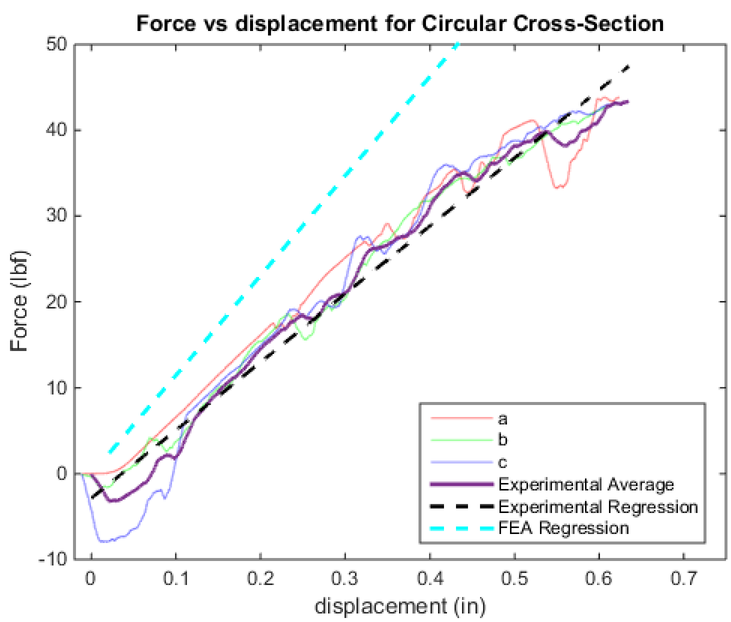


Figure 29: Force vs. Displacement Regressions and Data

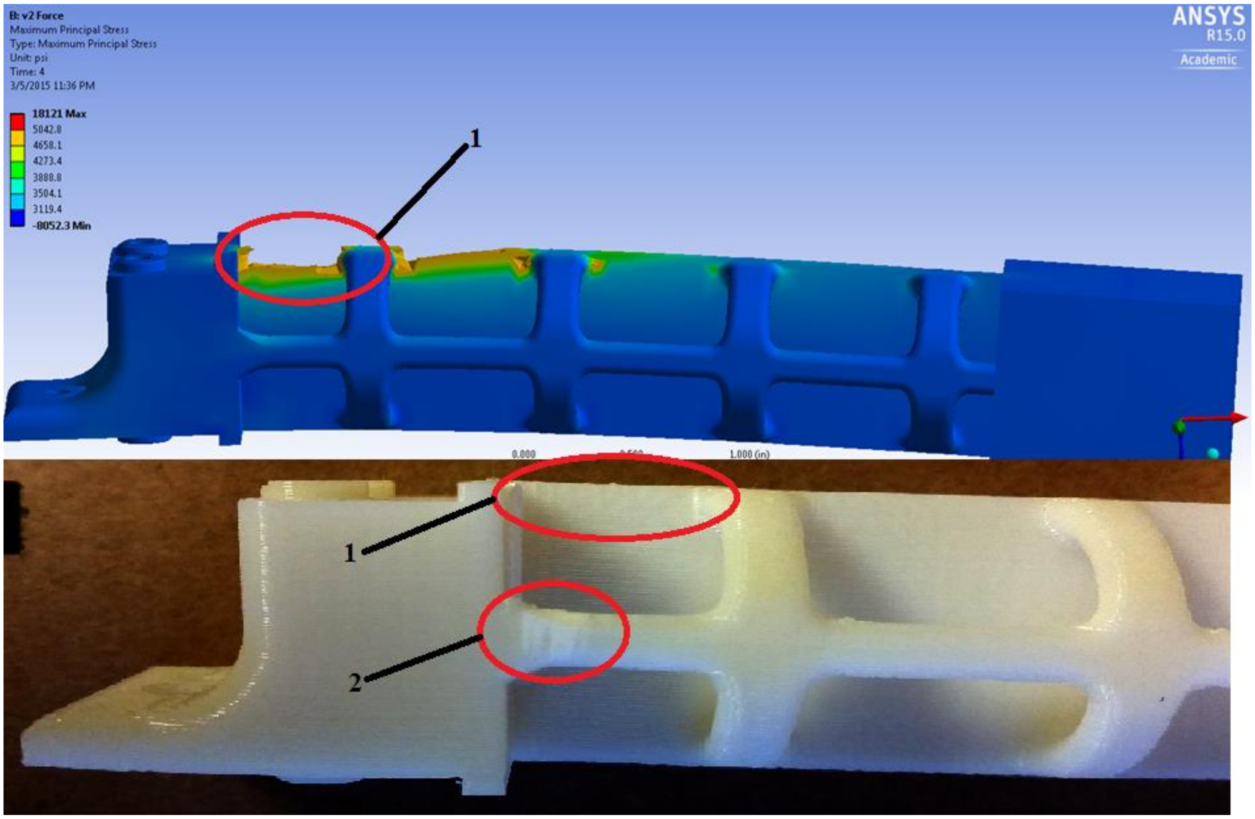
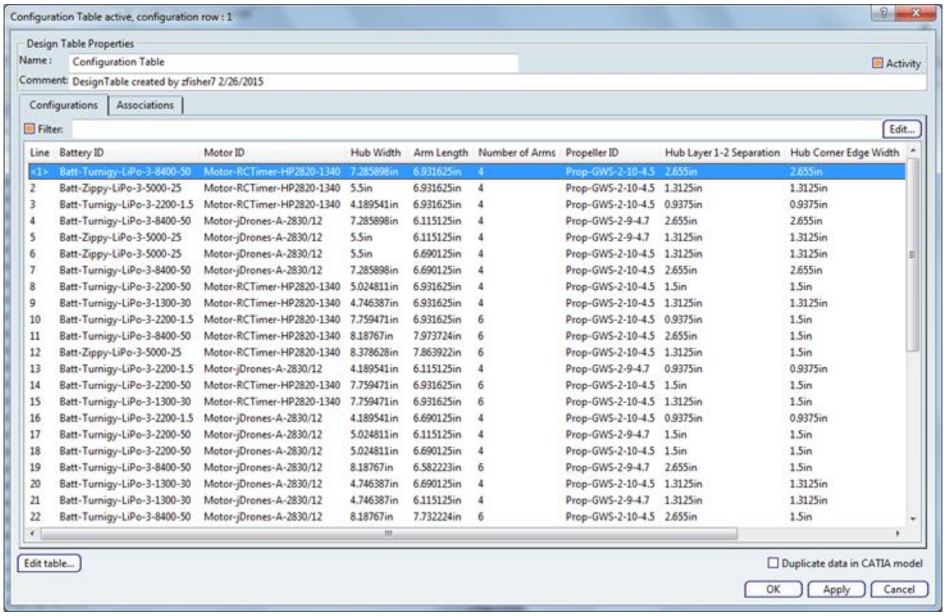


Figure 30: Complex Part Analysis

Using these verified models, the team built a conceptual and preliminary modeling environment containing all of the elements shown in Figure 16 up to, but not including the physical model. To pass the information from the conceptual and preliminary modeling to the physical models, Microsoft Excel tables were created which contained key information about the design that would drive the skeleton model and the selection of components. These tables were ranked using TOPSIS and allowed the user to select a feasible design that best met their needs. Figure 31 shows an example of the information read into the 3D modeling environment from Microsoft Excel. For the multicopter design, models for different batteries and motors were stored in separate part files within CATIA. Once a design was selected from the table shown in Figure 31, the parts and gross dimensions determined by the preliminary and conceptual modeling were used to update the 3D model.



Line	Battery ID	Motor ID	Hub Width	Arm Length	Number of Arms	Propeller ID	Hub Layer 1-2 Separation	Hub Corner Edge Width
<1>	Batt-Turnigy-LiPo-3-8400-50	Motor-RCTimer-HP2820-1340	7.285898in	6.931625in	4	Prop-GWS-2-10-4.5	2.655in	2.655in
2	Batt-Zippy-LiPo-3-5000-25	Motor-RCTimer-HP2820-1340	5.5in	6.931625in	4	Prop-GWS-2-10-4.5	1.3125in	1.3125in
3	Batt-Turnigy-LiPo-3-2200-1.5	Motor-RCTimer-HP2820-1340	4.189541in	6.931625in	4	Prop-GWS-2-10-4.5	0.9375in	0.9375in
4	Batt-Turnigy-LiPo-3-8400-50	Motor-jDrones-A-2830/12	7.285898in	6.115125in	4	Prop-GWS-2-9-4.7	2.655in	2.655in
5	Batt-Zippy-LiPo-3-5000-25	Motor-jDrones-A-2830/12	5.5in	6.115125in	4	Prop-GWS-2-9-4.7	1.3125in	1.3125in
6	Batt-Zippy-LiPo-3-5000-25	Motor-jDrones-A-2830/12	5.5in	6.690125in	4	Prop-GWS-2-10-4.5	1.3125in	1.3125in
7	Batt-Turnigy-LiPo-3-8400-50	Motor-jDrones-A-2830/12	7.285898in	6.690125in	4	Prop-GWS-2-10-4.5	2.655in	2.655in
8	Batt-Turnigy-LiPo-3-2200-50	Motor-RCTimer-HP2820-1340	5.024811in	6.931625in	4	Prop-GWS-2-10-4.5	1.5in	1.5in
9	Batt-Turnigy-LiPo-3-1300-30	Motor-RCTimer-HP2820-1340	4.746387in	6.931625in	4	Prop-GWS-2-10-4.5	1.3125in	1.3125in
10	Batt-Turnigy-LiPo-3-2200-1.5	Motor-RCTimer-HP2820-1340	7.759471in	6.931625in	6	Prop-GWS-2-10-4.5	0.9375in	1.5in
11	Batt-Turnigy-LiPo-3-8400-50	Motor-RCTimer-HP2820-1340	8.18767in	7.973724in	6	Prop-GWS-2-10-4.5	2.655in	1.5in
12	Batt-Zippy-LiPo-3-5000-25	Motor-RCTimer-HP2820-1340	8.378628in	7.863922in	6	Prop-GWS-2-10-4.5	1.3125in	1.5in
13	Batt-Turnigy-LiPo-3-2200-1.5	Motor-jDrones-A-2830/12	4.189541in	6.115125in	4	Prop-GWS-2-9-4.7	0.9375in	0.9375in
14	Batt-Turnigy-LiPo-3-2200-50	Motor-RCTimer-HP2820-1340	7.759471in	6.931625in	6	Prop-GWS-2-10-4.5	1.5in	1.5in
15	Batt-Turnigy-LiPo-3-1300-30	Motor-RCTimer-HP2820-1340	7.759471in	6.931625in	6	Prop-GWS-2-10-4.5	1.3125in	1.5in
16	Batt-Turnigy-LiPo-3-2200-1.5	Motor-jDrones-A-2830/12	4.189541in	6.690125in	4	Prop-GWS-2-10-4.5	0.9375in	0.9375in
17	Batt-Turnigy-LiPo-3-2200-50	Motor-jDrones-A-2830/12	5.024811in	6.115125in	4	Prop-GWS-2-9-4.7	1.5in	1.5in
18	Batt-Turnigy-LiPo-3-2200-50	Motor-jDrones-A-2830/12	5.024811in	6.690125in	4	Prop-GWS-2-10-4.5	1.5in	1.5in
19	Batt-Turnigy-LiPo-3-8400-50	Motor-jDrones-A-2830/12	8.18767in	6.582223in	6	Prop-GWS-2-9-4.7	2.655in	1.5in
20	Batt-Turnigy-LiPo-3-1300-30	Motor-jDrones-A-2830/12	4.746387in	6.690125in	4	Prop-GWS-2-10-4.5	1.3125in	1.3125in
21	Batt-Turnigy-LiPo-3-1300-30	Motor-jDrones-A-2830/12	4.746387in	6.115125in	4	Prop-GWS-2-9-4.7	1.3125in	1.3125in
22	Batt-Turnigy-LiPo-3-8400-50	Motor-jDrones-A-2830/12	8.18767in	7.732224in	6	Prop-GWS-2-10-4.5	2.655in	1.5in

Figure 31: Ranked List of Feasible Designs

The final step in the design modeling piece was to automate the detailed design. At a high level, this boils down to translating high level geometry and design parameters to detailed geometry that can be sent directly to an automated manufacturing process like a 3D printer. Like traditional design, 3D CAD files were created to detail the final design. However, the automation was handled using two main types of tools: a skeleton model and logical rules.

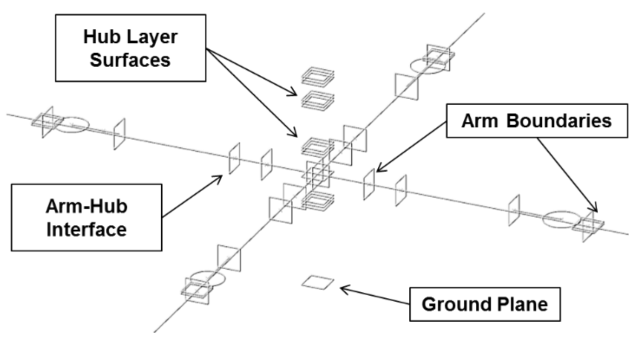


Figure 32: Multicopter Skeleton Model

Figure 32 shows the skeleton model for the multicopter vehicle. A skeleton model is really the physical manifestation of the product platform and the key elements identified are highlighted in Figure 32. The key physical features that are driven by conceptual design tools have been highlighted. The skeleton also handled the interfaces that were not addressed in the conceptual design tools. Ultimately, the skeleton was the link that lets design parameters be mapped to physical geometry. Another critical function of the skeleton model was to eliminate circular geometry dependencies, making dependencies unidirectional. A skeleton model removes many of the circular dependencies as all parts are built onto the skeleton, not each other. If the design changes, the skeleton first changed and then subsequent geometry is rebuilt onto the skeleton and only the skeleton. The first element updated after the conceptual dimensions have been passed to the CAD package was the skeleton model, and then each of the part files placed their respective geometry on this skeleton.

The second piece to automating the detailed design process was coding in the logic for the detailed design. A standard detailed design process must be completed for each part in the vehicle, with careful attention to the dimensions that will be left modular. For example, the analysis for the wall thickness of the multicopter arms was conducted on the range of potential arms. The output of this analysis (which was driven by the validated models) was not a single wall thickness, but a formula for wall thickness which is dependent on the length and the forces the arm is expected to carry. These forces are dependent on the weight, motors, props, etc. Multiple of these detailed part level analyses were encoded to successfully automate the part design. Some of these analysis were coded in the toolset outside the CAD package and the results were passed in via an excel table. Other analysis were coded directly in the 3D CAD software. Both strategies were successfully implemented, and the team recommends using which ever may be easier for the future user.

In addition to the part design, configuration level logic had to be encoded. Figure 33 shows a stored multicopter design and Figure 34 shows an updated design in response to a change in requirements. A number of key differences driven by the encoding of design logic been highlighted in Figure 34. The simplest change is that the motors and propellers have been swapped in the new design. One highly representative change is that the internal layout of the parts has been updated. This required algorithms for efficiently packing parts and wiring, and in this case study a hub packing algorithm was created to efficiently pack the components into the hub of the multicopter. However, in many cases the parts could not fit on a hub which met the span constraints. As a result, new layers to the hub assembly had to be created. This type of update required the encoding of which patterns will be repeated when expansion is necessary. The update of the number of hubs leads to a requirement for new bolts. For this type of change, standard libraries of engineering parts, such as the table of bolts, were created and the appropriate bolt could be selected. These standard elements of the component library could be shared across projects beyond the product family being designed, and it is the MASR research team's recommendation that further research into how to best manage an organization wide component library (where only certain elements are checked out for each project) should be conducted as future work.

Once the updates to the vehicle design have been made, a series of checks were encoded into the automated design environment. These include manufacturing checks such as the following examples: fillets to conform to the requirements for 3D printing, tool spacing for assembly, propeller clearances, etc. For this case study, the conceptual and preliminary design modeling environment was not linked to the detailed design environment in a way that could allow for iterative optimization. As a result, design properties like the center of gravity and total mass had to be checked against the ones predicted by the conceptual tools to ensure that they were within tolerance. Ideally this linkage would have been created allowing for better optimized vehicles, rather than those that fell with a tolerance of goodness.

Figure 33: Stored Multicopter Design

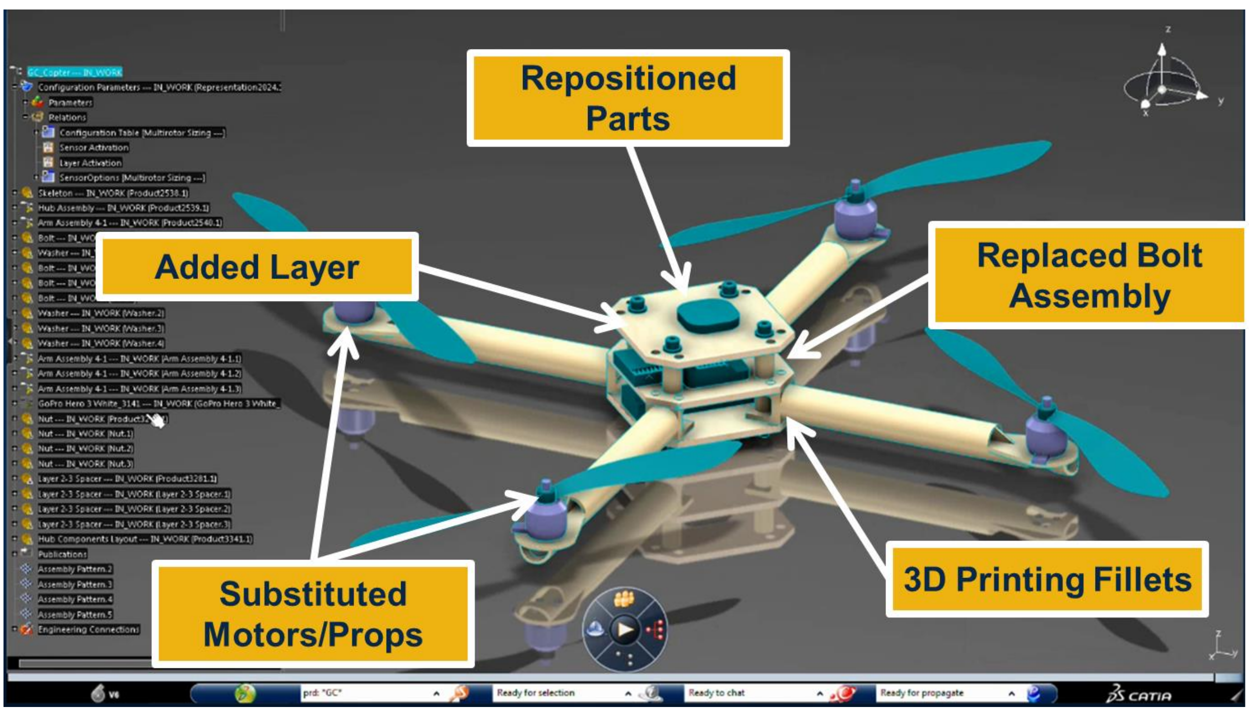


Figure 34: Automatically Generated Multicopter Design

With these models complete, the design performance could be simulated to ensure that it met the mission requirements. For the designs and missions selected, the modeling that was part of the conceptual design served as a reasonable mission simulation model and no additional simulation was necessary. For more complex missions, a simulation such as that performed in the indoor mapping scenario in Phase I of the MASR project may be required.

Once a confirmation of the designs ability to fulfil the requirements has been completed, the parts are exported in a manner that allows for manufacture. In the case study, stereolithography (.stl) files were exported directly from the CAD files for use in the additive manufacturing tools. 2D engineering drawing files were automatically generated for use in the laser cutter.

The ADAPt design process provided an approach for automating the design of new elements within a multi-platform product family. The approach presented is described in a strictly linear manner due to the constraints of a written document, but like any design, it was implemented in a highly iterative fashion.

USER INTERFACE

While the ADAPt design process allowed the engineering process to be automated, to fulfil the need of the soldier, this automated design had to be paired with a simplified user interface. Figure 35 shows the user interface developed. This interface allowed the user to enter requirements and constraints which would feed the ADAPt Design process as a back end.

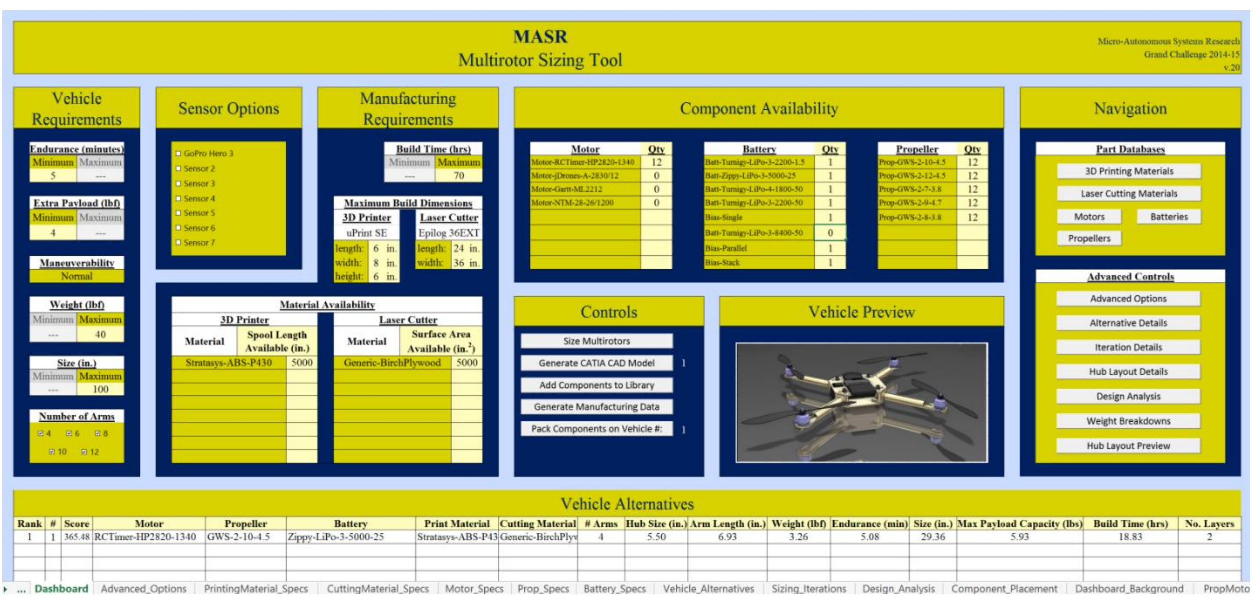


Figure 35: MASR Multirotor User Interface

The left column in Figure 35 contains a section for vehicle requirements to be entered by a soldier. The box at the top left of Figure 35 allows the selection of a series of commonly used and pre-determined sensors. The box next to that allows the user to enter the manufacturing requirements, such as allowable manufacture time, tooling availability and bed size, material machining capabilities, etc. These element then become constraints in the ADAPt Design process. The top center box in Figure 35 allows the user to enter component availabilities. These also constrain the design. The right column in Figure 35 provides a series of buttons that allowed the user to access the differing pages of the component database. The bottom section of the user interface was reserved for displaying the ranked list of feasible designs along with their performance characteristics such as payload and endurance.

This user interface combined with the developed ADAPt process satisfied the needs listed in Figure 4. The next section is dedicated to the ARL-Georgia Tech team's testing of this approach.

TESTING AND RESULTS

This section details the efforts the team made to test the approach developed. Because the approach was developed in conjunction with elements of the ADAPt Design process, multiple tests were used on differing aspects of the design process to ensure the approach was applicable beyond the problem for which it was developed.

TESTING OPERATIONAL RELAVANCE

As a mechanism to test the operational relevance of the approach, the ARL and Georgia Tech team worked to identify a potential scenario where micro-autonomous systems may improve situational awareness. The following scenario was intended to provide a realistic test of the developed approach and toolset.

Figure 36: Platoon Level Mission

To set the stage for this test, it is necessary to move the perspective to the scale relevant to the platoon and squad. Figure 36 shows a combat area in which there is a city with a complex urban terrain split by a river. There are a limited number of bridges crossing this river.

In this scenario, the urban terrain surrounding the bridge can be assumed to be hostile. Figure 37 shows a local view of the bridge, where a platoon of soldiers is expected to make semi-regular patrols through the potentially hostile city. The terrain around the bridge allows for clear sight of the bridge from multiple overlooking buildings and reasonable access to the bridge from either the land or water. Prior to crossing the bridge, the platoon must inspect underside of bridge decking and structural supports, and interrogate small obstacles and trash on the roadway for explosives. This puts soldiers on foot at risk due to the overlooking building. The bridge crossings can be expected to continue with some regularity for a few months, which is enough time for both friend and foe to assess the situation, but not enough time to receive materiel tailored to the scenario via traditional acquisition processes. After a few months securing the area, the platoon will redeploy and will likely face new unforeseeable highly localized challenges.

Figure 37: Local View of Contested Bridge

Figure 37 highlights some of the key features of the bridge. The soldier examining the bridge can determine and set a number of requirements based on the local needs. The length of the bridge is 600 ft., and it normally takes a platoon 10 minutes to fully inspect it. The bridge has constricted/cramped features that necessitate a high maneuverability and a small size. Moreover, in order to make the inspection possible, the UAV assets must have hover capability. The asset must be easily carried without adding an excessive additional weight to the soldiers. In order to detect explosive devices, a live video feed is needed.

These locally driven requirements can be summarized as follows:

- **Bridge Inspection Mission Requirements**
    - Endurance: 10 minutes
    - Maneuverability: High, with hover capability
    - Maximum Size: 33 in
    - Maximum weight: 6 lbf
    - Sensors**:** Live Video Feed

In the described scenario, the soldier identifies that a Micro-Autonomous Aerial vehicle may provide him a solution to the bridge inspection that reduces his personal exposure.

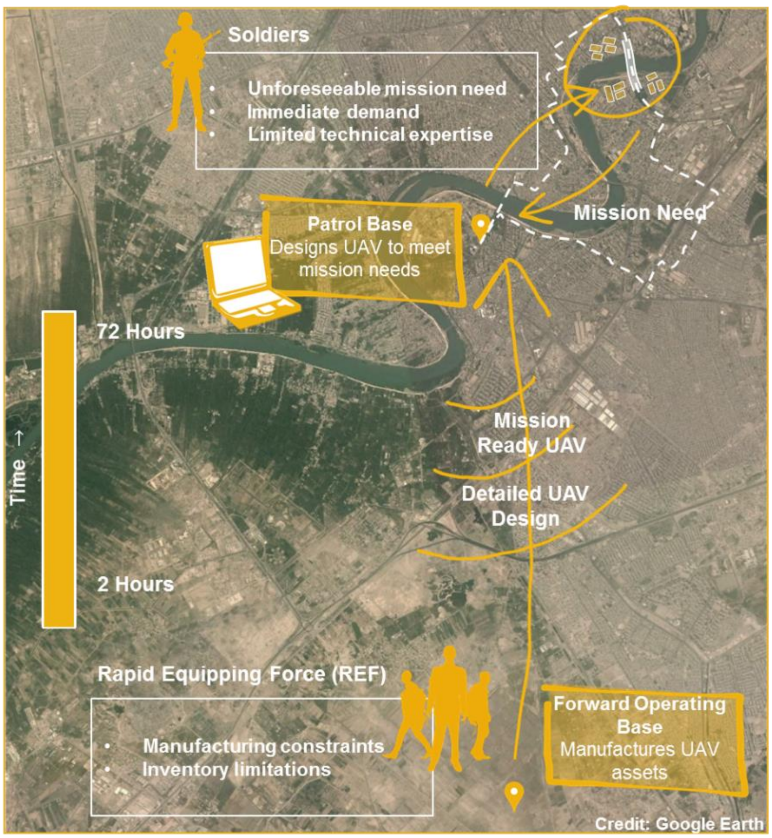


Figure 38: Concept of Operations

For our bridge scenario, the concept of operations for an "on-demand" model is shown in Figure 38 and would work as follows:

The platoon conducts a recon route one day and notices the bridge, and that it is surrounded by urban terrain including buildings. They realize that it would be beneficial to have a UAV with a live video feed that could be deployed from the safety of their Humvee to conduct the bridge inspection rather than having to get out and inspect the bridge on foot. They note that the openings on the bridge are 33 in. and that it took them 10 minutes to inspect it the first time.

The platoon returns from their route at the end of the day to the patrol base and utilizes a computer terminal loaded with UAV design software described above has been loaded. They generate a design to satisfy the mission requirements listed. Then, detailed drawings and manufacturing files are sent to an operating base where the rapid equipping force REF is stationed. The REF operates mobile manufacturing labs equipped with 3D printers, laser cutters, and CNC mills. The REF takes the design sent to them and turns it into a mission ready UAV. In less than 3 days (prior to the next expected patrol), the UAV is shipped back or flown back autonomously where it can be used to inspect the bridge.

To simulate the soldier's needs, the requirements listed in Table 2 were entered into the interface shown in Figure 35.

Table 2: Soldier Requirements

Requirement	Value
Max Outer Dimension (in)	33.0
Max Weight (lbf)	6.0
Min Endurance (min)	10.0
Max Build Time (hrs)	30.0
Sensor	GoPro

Table 3 shows the design created to meet the mission needs. A physical vehicle was created using additive manufacturing and a laser cutter via the process described. This vehicle can be seen in Figure 39, and the actual dimension, weight, endurance, and build time are listed next to the values predicted by the modeling environment. The dimension and weight are very close to the values predicted by the modeling environment. The real vehicle outperformed expectations by %19.9. Some investigation revealed that the battery was capable of holding more capacity than the specification for the battery being used stated. The build time also differed from the predicted build time due to a difference in expected worker assembly speed and realized worker assembly speed.

Table 3: Tailored Design as Predicted by Modeling and as Realized

Metric	Predicted	Actual	Error
Outer Dimension (in)	29.7	29.7	0.0%
Weight (lbf)	3.08	2.99	3.0%
Endurance (min)	12.1	15.1	19.9%
Build Time (hrs)	18.0	20.5	12.2%

Figure 39: Tailored Design for Bridge Inspection

Figure 40 shows two additional designs automatically created by the modeling environment for two different missions. The first is a similar recon mission, which leads to the design listed. The second is a communications relay mission where the multicopter is expected to hover with a relatively heavy payload while a message is transferred. The requirements for this mission led to the necessity of a design with six propellers for improved payload capacity.

Short Range Recon Mission

- Portable (small, light)
- High endurance
- Imaging capability

Requirement	Value	UAV Design
Max Outer Dimension (in)	33.0	26.7
Max Weight (lbf)	10.0	2.98
Min Endurance (min)	10.0	16.2
Max Build Time (hrs)	22.0	16.1
Extra Payload (lbs)	0.0	0.99
Sensor	GoPro	

Notional communications relay mission

- Hover with 8 lb. payload

Requirement	Value	UAV Design
Max Outer Dimension (in)	50.0	34.1
Max Weight (lbf)	20.0	5.16
Min Endurance (min)	7.0	7.18
Max Build Time (hrs)	40.	33.1
Extra Payload (lbs)	4.0	8.63
Sensor	none	

Figure 40: Additional UAV Designs and their Parameters

TESTING ALTERNATIVE DESIGNS AND TOOLSETS

The "on-demand" concept of operations and the ADAPt Design process were developed in unison with the development of the multicopter vehicle class. This design process relied heavily on data driven models and off-the-shelf components. The simplicity of the vehicle allowed this particular approach. The design was modeled largely in Microsoft Excel and CATIA V6.

To test the relevance of the process outside the concept on which it was created a second class of fixed wing vehicles was implemented using a drastically different toolset, but the same process. This section presents the research conducted to automate the manufacturing of a 3D-printed aircraft.

REQUIREMENTS ANALYSIS AND ARCHITECTURE SELECTION

For the development of the fixed wing design, the requirements analysis and much of the architecture selection phases could be borrowed from the original analysis presented. Since the fixed wing design had originally been intended to be part of the multi-platform product family, the steps presented in the approach section were still valid. Small modifications were made to the component library to include a few different item, such as higher speed propellers. These items were kept separate from the propellers in the multicopter library because the data stored about them differed (differences are described in the modeling section). The initial platform skeleton and back of the envelope design for the fixed wing vehicle was conducted in SolidWorks. The team found this CAD package lacked the same level of top down design support that some of the other packages provided. As a result, the fixed wing design aircraft's skeleton was managed externally to the CAD software. However, the same principals of laying out the key interfaces and elements on a skeleton that reduces iterative geometry feedback loops was applied.

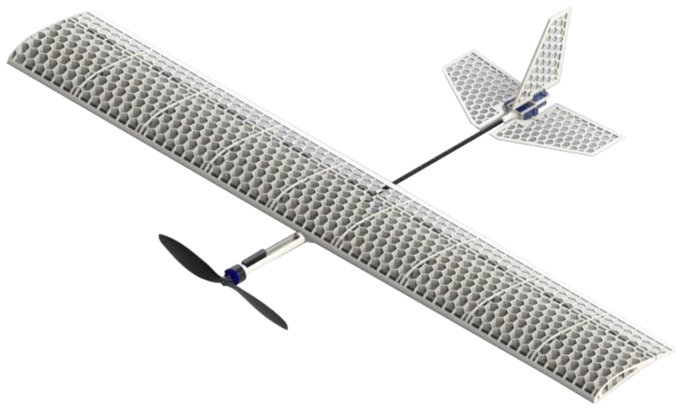


Figure 41: Early Fixed Wing Design

Figure 41 shows an early version of the fixed wing aircraft. The aircraft was built around a set of carbon rods that created a cross and remained relatively constant throughout the design process.

INTERFACE DESIGN

The overall design goal was to produce a fixed wing RC aircraft that could be easily modified, manufactured, and assembled depending on mission requirements. This design goal was the primary driver for all design decisions made. The aircraft was designed to allow and aircraft on-demand with minimal expertise required. This "select, print and fly" design was achieved in two parts. First, all 3D printed parts were designed to be modular as well as easily scalable. Second, all parts and materials that could not be 3d printed were chosen based on their off the shelf availability.

The following sections detail the choices made in interface definition. Since the design was built around a rod, the vast majority of the design consisted of either the wing or attachments to allow elements to easily attach to the rod.

WING

The wing airfoil selection was deemed to be a discrete choice. In the final design, three airfoils were selected as potential discrete options. To enable this, the section representation was standardized. Each section was represented by a common number of points which could be read into the CAD packages as a new airfoil section. The team explored differing CAD packages and found both SolidWorks and CATIA allowed this process to be automated with relative ease. For CREO Parametric 3.0 used in the fixed wing design, it was easier to pre-select airfoils and create a sketch for each section, limiting the flexibility of the model. The next section will discuss the three sections chosen.

AIRFOIL SELECTION

Three airfoil types were selected to give a large range of performance characteristics and design potential. The selected airfoils were the NACA0012, Clark Y, and Selig1223. The NACA0012 was chosen as because of its simplicity. It was used to verify that the 3d printer was cable of printing airfoil sections with sufficient accuracy and consistency while also providing an airfoil that allows for high maneuverability characteristics.

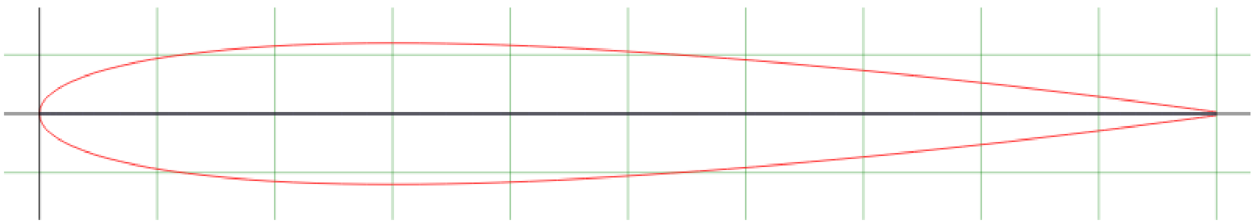


Figure 42: NACA0012 Airfoil

The Selig1223 airfoil was chosen because it would provide the high lift at low Reynolds numbers. The Selig1223 also represents a conventionally difficult to manufacture shape due to its high camber, that could be made more efficiently using a 3d printer. A high lift airfoil was necessary because early projections for the aircraft's weight were higher than that of traditional RC planes of a similar size.

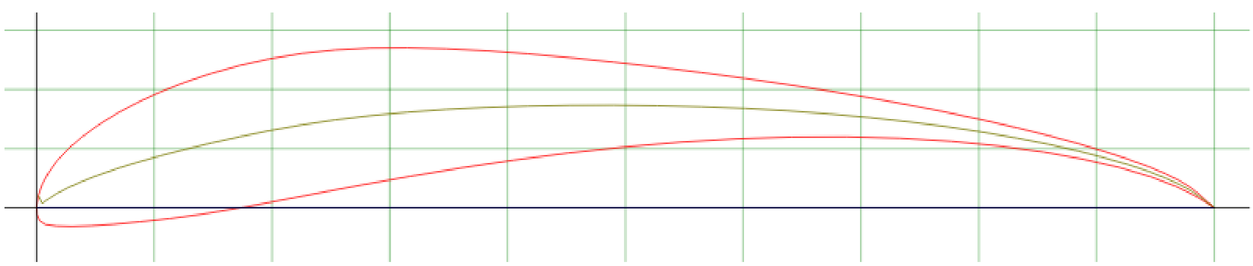


Figure 43: Selig S1223 Airfoil

The Clark Y was chosen because it is a well-tested and proven airfoil on RC planes. The Clark Y is also a semi symmetrical airfoil with a non-excessive camber which places it nicely between the symmetric NACA0012 and highly cambered Selig1223 airfoils.

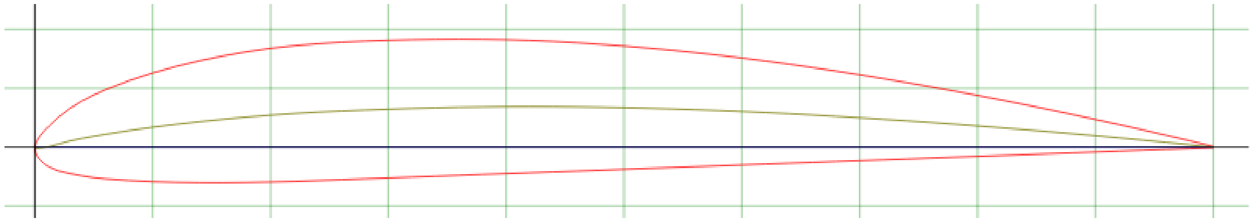


Figure 44: Clark Y Airfoil

WING SECTION GEOMETRIC FEATURES

The aircraft's wing had to be printed in multiple wing sections due to the limitations of the 3D printer build area. The maximum wing section size was set to a 7.5 inch chord and 5.5 inch span. It was determined that each section would sit adjacent to each other on the wing spar, and the skin (packing tape) would be used to keep the sections in place.

The internal structure of the wing section was designed as a honeycomb fill. This was done to provide a high amount of bending and torsional stiffness while still keeping the weight low. The diameter and spacing of the honeycomb pattern was chosen after several revisions. The goal while adjusting the honeycomb pattern was to reduce the part print time and mass while still maintaining sufficient stiffness. Sufficient stiffness was based on preliminary testing of prototype parts.

Finally, many of the holes required a running fit to allow free rotation. To accomplish the running fit holes were intentionally undersized and then manually drilled to the appropriate size to maintain tighter tolerances than the 3d printing process allowed. This process was especially necessary in the rear spar interface for the wing sections shown at the back of Figure 45.

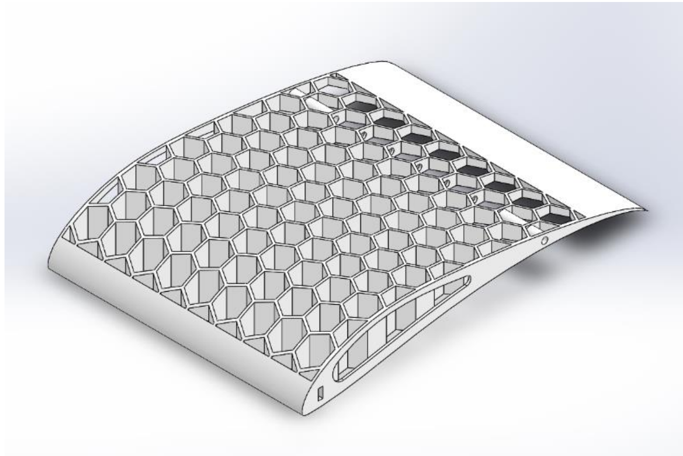


Figure 45: Selig S1223 Airfoil Section

EMPENNAGE

The tail assembly was designed as a stabilator, or all-moving tail. The all moving tail was chosen because it allows for the empennage to control pitch, yaw, and roll. Controlling all principal axes from the tail allows for the control surfaces to be removed from the wing. Removing the control surfaces from the wing allowed for large reduction in design complexity and number of interfaces with a small trade-off in aircraft maneuverability. The removal of control surfaces from the wing also allowed for a weight reduction through the removal of servomotors necessary for controlling the ailerons. All in all, it is a simpler design with fewer moving parts, although it is not traditional in RC-type aircraft.

The interface between the tail and the carbon fiber rod, that served as the fuselage, was a hole in which the carbon fiber rod would be placed with a set screw to lock the empennage in place. Figure 46 shows an early empennage design, and Figure 47 shows the final empennage design. Preliminary testing of the original empennage design at realistic airspeeds revealed that the initial design could not handle the range of forces at the interface between the servomotor and the control surfaces. The interface design was updated to contain a snap together 3D printed part that could fit around a standard servo horn. This particular example demonstrates the need to test the interfaces across the full spectrum of the design range.

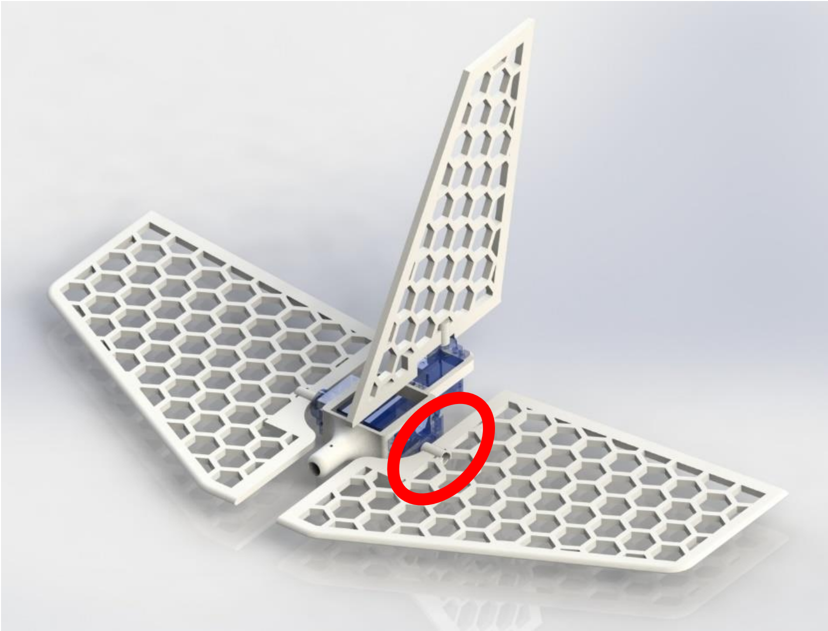


Figure 46: Early Empennage Design

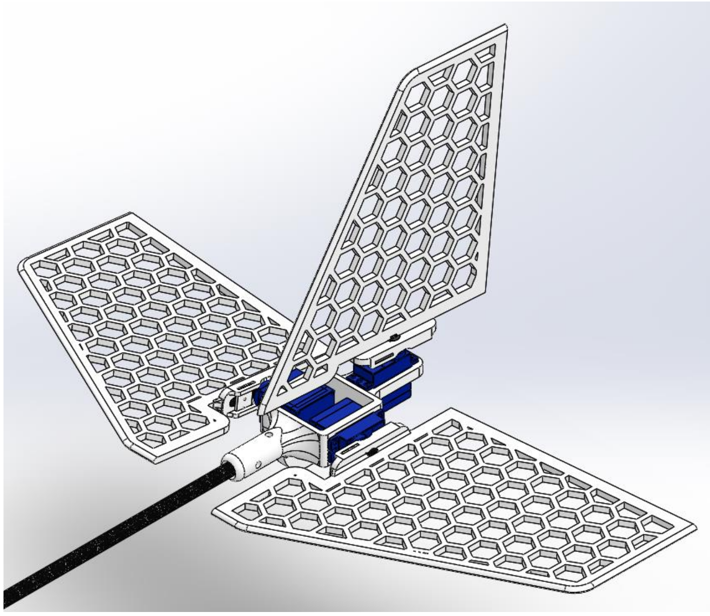


Figure 47: Empennage

FUSELAGE

The main component to the fuselage is a 0.25 inch carbon fiber rod. The carbon fiber rod was chosen to help minimize the weight of the aircraft. It was also used as the primary attachment for the aircrafts components. Using the carbon fiber rod as the primary attachment point allowed for the aircraft to remain modular and simple to assemble. This carbon fiber rod could be replaced by a locally sourced wooden dowel rod of larger diameter if necessary.

The primary interface between the component connectors and the fuselage tube, was the creation of a sleeve that fit over the rod. The details of each sleeve are discussed below in the discussion of the attachment of accessories.

MOTOR MOUNT

The motor mount was designed so that it could be easily extended or shortened depending on how the center of gravity needed to be shifted. The bottom of the motor mount also has a plate and hole pattern designed to allow the landing gear assembly to be easily mounted.

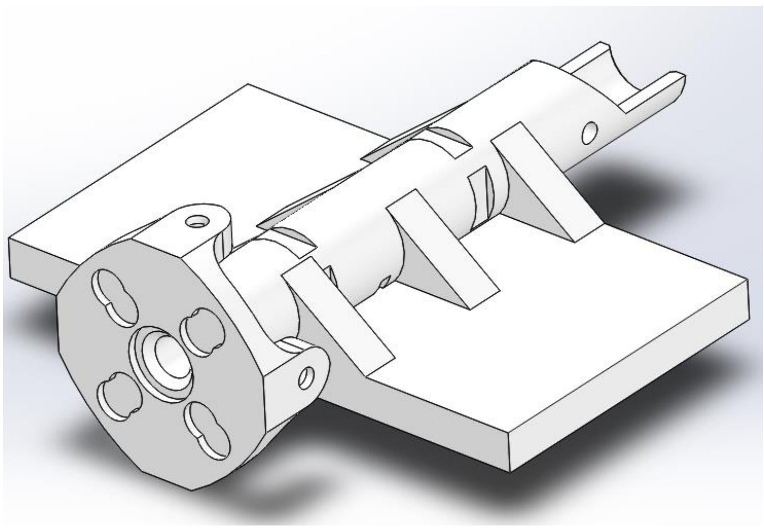


Figure 48: Motor Mount

ADAPTER PLATE

The adapter plate is located at the front of the motor mount. It is used as the attachment between the aircrafts motor and motor mount. The adapter plate was designed so that the motor hole patterns could be adjusted and changed without requiring the entire motor mount to be reprinted, reducing print time between different builds.

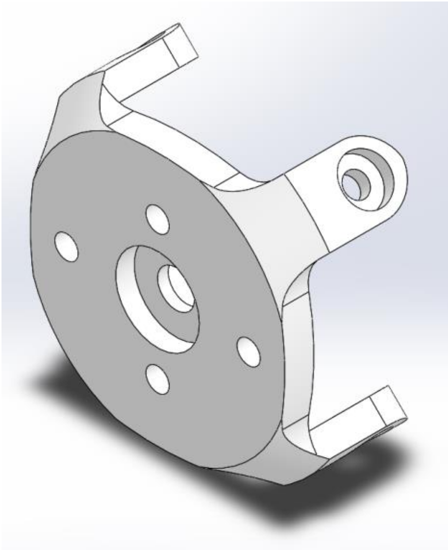


Figure 49: Adapter Plate

LANDING GEAR ASSEMBLY

The landing gear assembly was built using off the shelf components. It uses a Du-Bro landing gear made from a single piece and designed to weigh less than an aluminum counterpart while being capable of absorbing landing impacts without bending. The wheels attached to the landing gear were sized so that the aircrafts propeller would have sufficient ground clearance on surfaces that are not completely flat.

Initially the team had constructed the design to be hand launched with skid landing, but the team determined a ground launched vehicle would improve operator safety and this landing gear was added.

ACCESSORIES

BATTERY CAGE

The battery cage was designed to protect the aircraft’s battery from hard impacts in the event of a crash. It was designed as a cage to keep the weight low while still being protective. The battery cage shown in Figure 50 also demonstrates a number of standardized features of the fuselage attachments discussed in the next section.

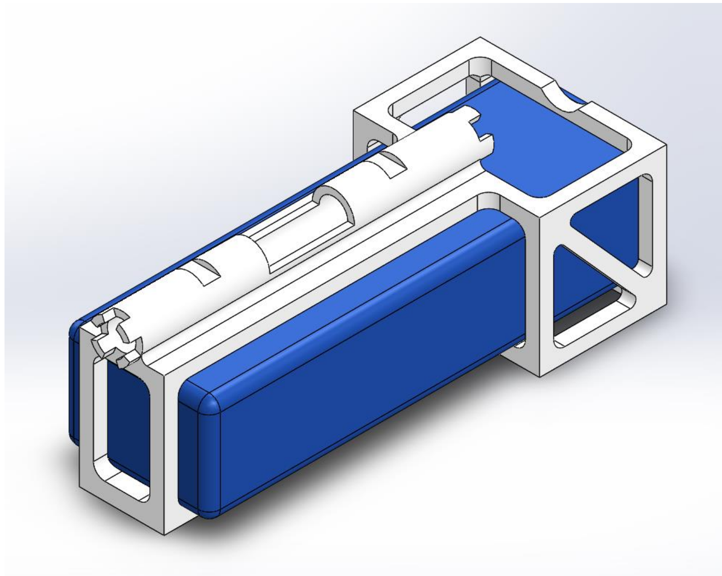


Figure 50: Battery Cage with Battery

ELECTRONICS MOUNTING PLATFORM

The electronics for the aircraft are mounted to flat mounting platforms. These mounting platforms, shown in Figure 51, were designed to be easily assembled and modular while allowing a large range of electronics. Each of the fuselage accessory attachments contained a number of standardized features. The first key standardized feature is the use of merlons that locked each element to the next. The mounting platform also has special slots designed to prevent zip ties from moving once strapped in place. The mounting platforms were designed with enough space that Velcro strips could be used to attach elements to these platforms. The electronic components were attached to these mounting plates with Velcro and zip ties.

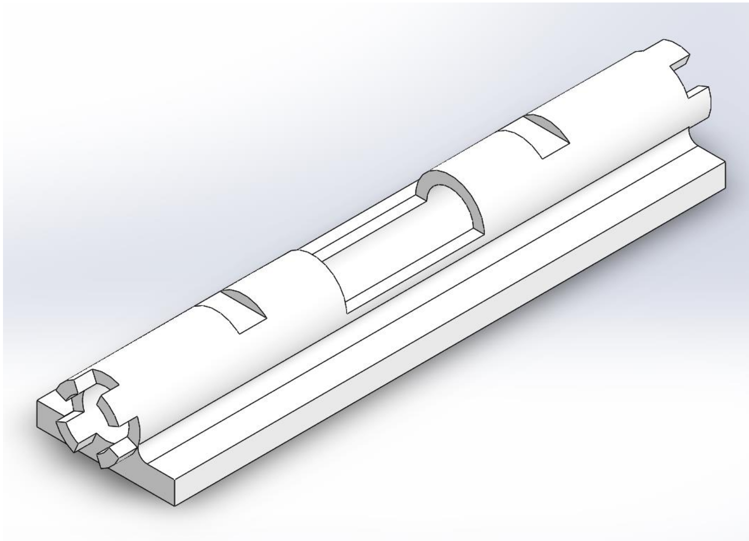


Figure 51: Electronics Mounting Platform

CONCEPT REFINEMENT AND DESIGN

CONSTRAINT IDETIFICATION

Many of the constraints on manufacturing processes and equipment availably could be transferred from the multicopter design. However, the 3D printer bed size limited each wing section to a 7.5 inch chord and 5.5 inch span. This constraint was found to severely limit the design freedom. In addition to this manufacturing constraint, it was determined that the possibility for hand launch of the UAV should be maintained. This added a minimum flight speed constraint for the moments after hand launch.

MODEL BASED DESIGN & DEVELOPMENT

The modeling environment for the fixed wing aircraft differed significantly from the multicopter in the way in which the models were constructed. For the fixed wing vehicle the requirement for test data to be collected within a wind tunnel shifted the team towards physics based modeling. In addition to the shift towards physical modeling, the continuous nature of the aircraft design space lead to an optimization based approach.

OPTIMIZATION SETUP & OBJECTIVE

The optimization objectives were the performance of the UAV (speed, endurance, maneuverability, payload capability) and the printing time. The constraints for the design are as follows: account for a set of discrete components with limited inventory, limited printing surface, and architecture constraints. This creates a constrained multi-objective optimization problem with both discrete and continuous variables that was solved using a NSGA-II algorithm.

CONCEPTUAL AND PRELIMINARY MODELING

Figure 52 shows an overview of the conceptual and preliminary modeling environment for the fixed wing aircraft. The next few sections will detail the elements of the model.

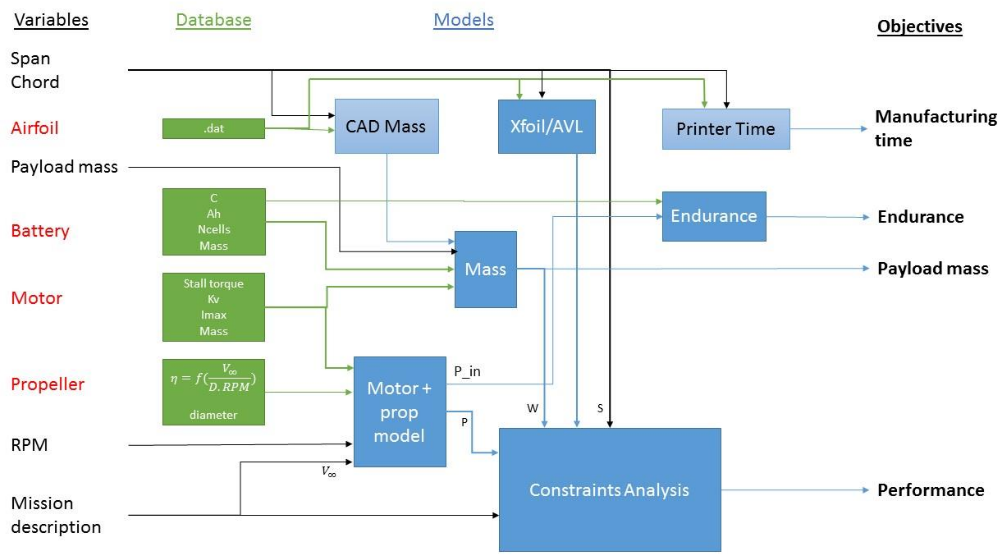


Figure 52: Modeling Overview

MODEL NOTATIONS

Metric system is used throughout the model.

V speed (m/s)

D propeller diameter (m)

n rotations per second

K_t

K_v

PROPULSION

The propulsion was made of a propeller, an electric motor, an ESC and a battery. There was a limited number of combinations that can work. Hence a full factorial was performed and a selections were made from there.

PROPELLER

Propeller are described by two parameters $A \times B$, where A is the diameter of the propeller, and B its pitch (both in inches). A propeller is characterized by its efficiency versus advance ratio curve. The advance ratio ('J') is a dimensionless parameter defined as follow:

$$J = \frac{V}{Dn}$$

A regression was performed on the data provided by (Ananda, 2015)[3]. A fourth order polynomial equation was used as a surrogate.

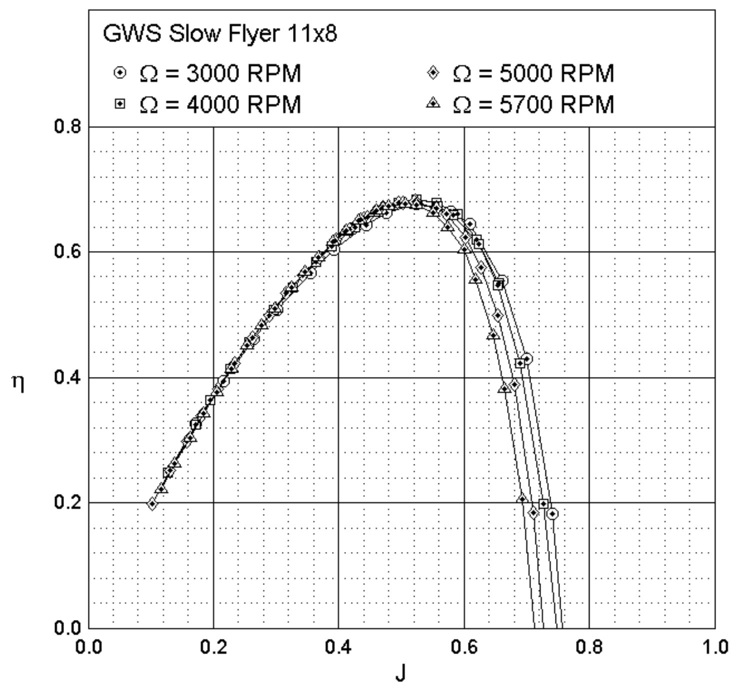


Figure 53: Propeller Test Data[3]

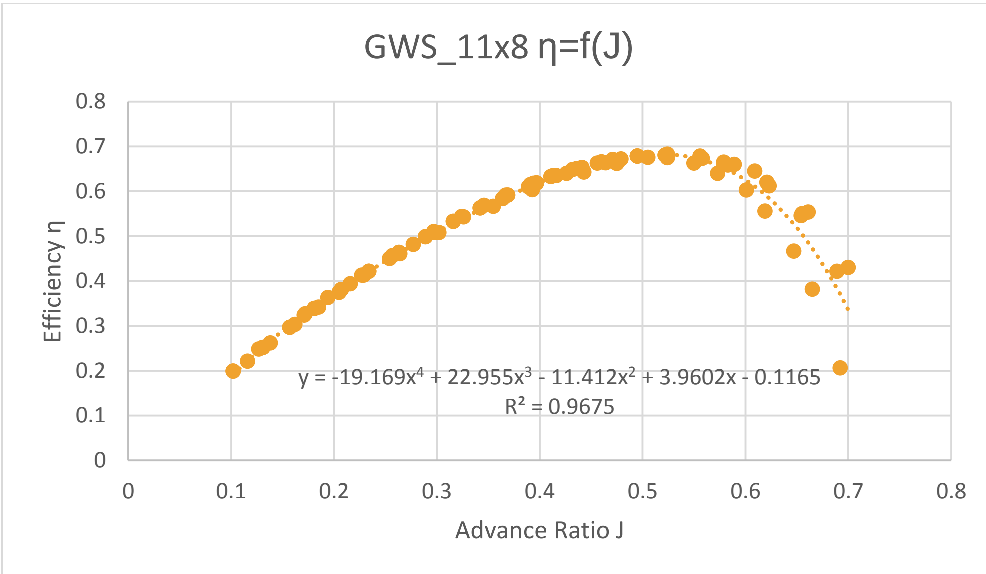


Figure 54: Regression of Propeller Efficiency

MOTOR

Brushless ESC controlled motors were used in the fixed wing design. The model built links the output power components (the product of rotational speed and torque) to the input power components (current and voltage).

The relationship between torque and motor speed was assumed to be linear.

In this model motor are characterized by their Kv, their max current and the slope of the torque/speed relationship.

$$K_t = \frac{30}{\pi K_v}$$

$$\mathrm{T} = K_t i$$

$$RPM_{max} = K_V * V_{in}$$

The slope of the torque/speed curve depends on the input power, approximated by the number of cells of the battery.

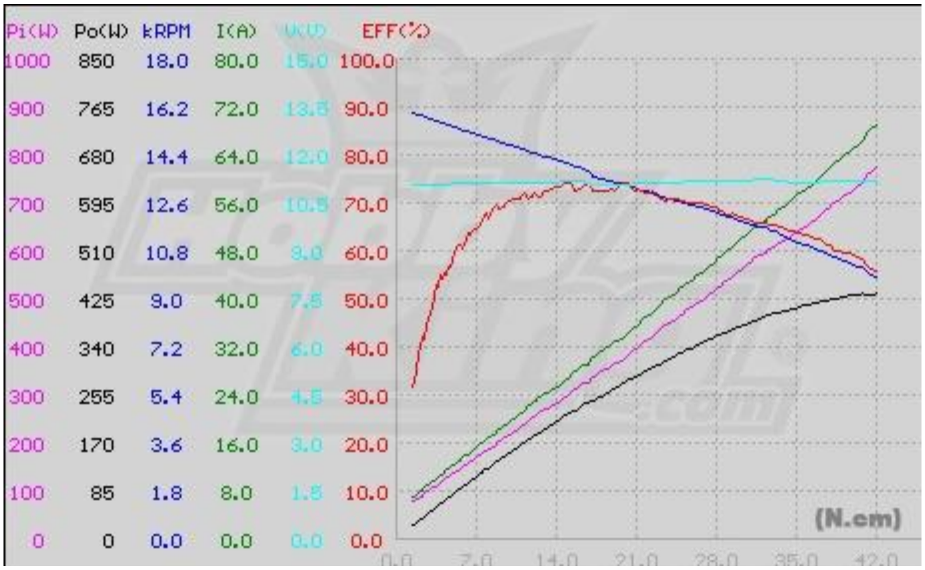


Figure 55: NTM dyno Test Data 3S LIPO (hobby King)

BATTERY

The fixed wing aircraft was powered by lithium polymer batteries that matched those in the component library for the multicopter.

The parameters of the battery were: the number of cells, their capacity rating and their capacity (in mAh). The number of cells impacts the voltage of the battery. The average voltage of a cell is around 3.7V, of course the actual value depends of the depth of discharge of the battery.

$$V = 3.7 * n_{cells}$$

The capacity of the battery indicates how much current can be drawn from the battery. For instance a 1 Ah fully-charged battery can deliver 1 A during 1 hour, or 2A during 30min, or 4A during 15min, before being empty.

The capacity rating indicates the maximum speed of discharge of the battery, i.e. the minimum time in which the battery can be discharged.

$$t_{min} = \frac{60}{C}$$

AERODYNAMIC ANALYSIS

XFOIL

XFoil is an open-source software (GPL license) developed by Pr. Mark Drela at MIT. It performs viscous analysis of airfoil. It was used in our model to compute the angle of attack for which there is the maximum section lift coefficient (angle of attack just before stall). It was also used to find the section drag coefficient at zero lift. Because of convergence issues at low Reynolds number (smaller than 500,000) a lower limit for the analysis has been set.

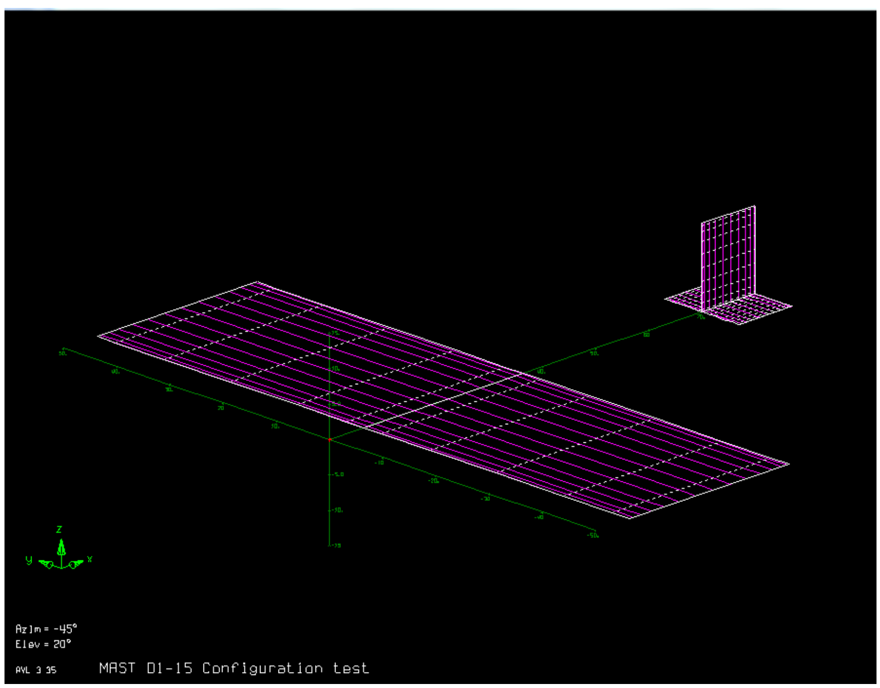
Many convergence issues were encountered in finding the zero lift drag for the Selig 1223 airfoil because of the very high camber.

Hence instead of computing those values at each call of the missionAnalysis code, the impact of the Reynold's number was neglected and the values assumed constant. The values were added to the excel spreadsheet.

$$C_{D_0} = C_{d_0}$$

AVL

AVL is an open-source (GPL license) software developed by Pr. Mark Drela at MIT. AVL is a program for the aerodynamic and flight-dynamic analysis of rigid aircraft of arbitrary configuration. It employs an extended vortex lattice model for the lifting surfaces. It performs non viscous analysis of the flow over the airplane. It is used to find the total lift coefficient of the aircraft before stall. This program takes a description of the plane as an input.



ENERGY BASED ANALYSIS

AERODYNAMIC MODEL

$$C_D = C_{D_0} + K_1 C_L^2 + K_2 C_L$$

$$K_1 = \frac{1}{\pi \mathrm{AR} e}$$

The value given by AVL for the efficiency factor was used when the results converged.

MASTER'S EQUATION

The energy balance between mechanical, potential and kinetic energy gives:

$$[T - (D + R)]V = W\frac{dh}{dt} + \frac{W}{2g_0}\frac{dV^2}{dt}$$

After a few transformations made by using the equations presented above the relationship between lift, weight and load factor ($nW = L$) one can get the master equation. The weight is constant throughout the flight (electric aircraft), the UAV is propeller driven, and the altitude is not high enough to have significant variation for the air density allowing the master equation presented below to be derived. Using this equation, the power to weight ratio and wing loading for each vehicle were determined within the conceptual design tool.

Master Equation

$$\frac{P}{W} = [\frac{K_1 . n^2}{q} . \left(\frac{W}{S}\right) + q . \frac{C_{D_0}}{\left(\frac{W}{S}\right)} + \frac{R}{W} + \frac{1}{V} . \frac{d}{dt}\left(h + \frac{V^2}{2g_0}\right)]$$

WING MANUFACTURING ANALYSIS

The Selig S1223 wing sections print time was tested against honeycomb diameter. The print time was also tested against part width. The results of this testing allowed the reduction in the print time, along with a model for the print time for the vehicle. The results of this testing are presented in **Error! Reference source not found.**.

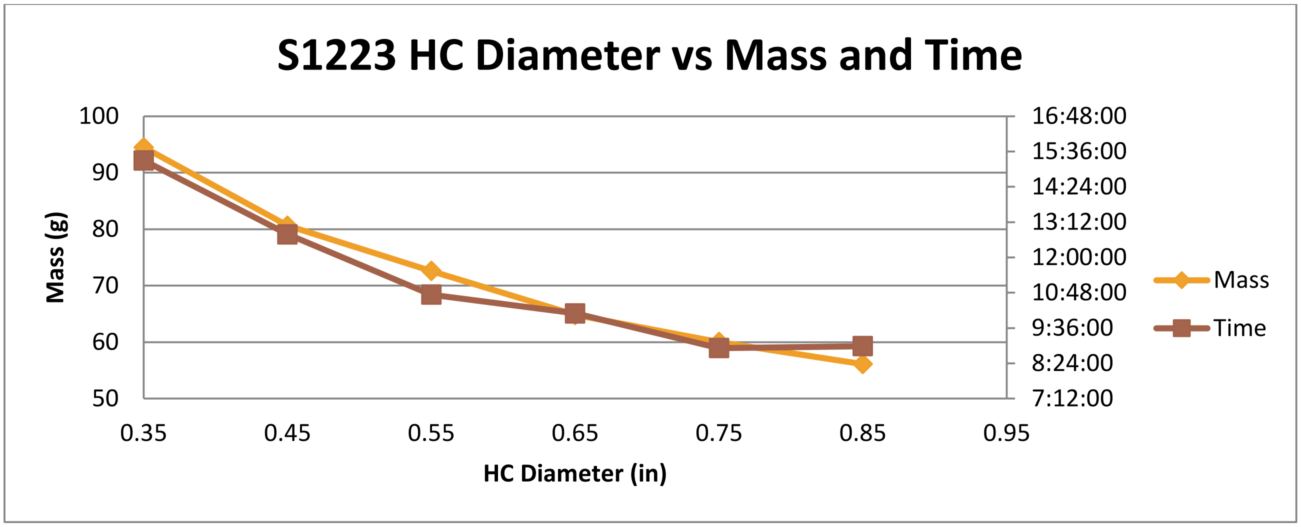


Figure 56: S1223 HC Diameter vs Mass and Time

MODEL INTEGRATION

The above elements were linked together as shown in Figure 52 and this modeling environment was optimized in Matlab using the NSGA-II Pareto frontier finding algorithm. Once the optimization was completed, it is necessary to help the user understand how their preferences relate to the differing Pareto optimal designs.

PARETO EXPLORATION

EXPANDING THE PARETO FRONTIER

With a single Pareto frontier, there was little variation across endurance and payload. Additional battery and motor options were added in an attempt to expand the design space. Constraints were also tweaked across multiple runs, and the results compiled into a wider Pareto frontier. The additional constraint cases are shown in Table 4, and the constrained bounds are shown in Table 5.

Table 4: Fixed Wing Design Requirements Matrix

	Endurance (min)			
Payload (g)		5	7.5	10
	100			
	200			
	300			

Table 5: Fixed Wing Design Input Ranges

Variable	Span (mm)	Chord (mm)	Payload (kg)	Speed (m/s)	RoC (m/s)	Load ('g')	Accel (m/s)	Launch Speed (m/s)
Lower bound	300	.15	.05	6	.5	1.05	.5	4
Upper bound	1210	7.5	1	25	5	2	5	8

Table 2 – Constraints for Simulation

There were no possible solutions found at 10 minutes of endurance for max payloads of 200 g or more. Duplicate solutions were eliminated from the frontier, and we ended up with 47 solutions.

FILTERING THE FRONTIER USING CASE REQUIREMENTS

To select one solution for a design, it was necessary to filter these results, setting minimums or maximums as deemed appropriate, and also set importance of each variable to a specific case. All solutions not meeting the requirements are eliminated, and the importance value is used to rank order the remaining solutions, and select one. The considered cases are shown in Table 3. These were designed in order to give us the most diverse solutions possible in order to test the capabilities of our parameterized model. Many other cases were examined, but these were selected for their versatility of results. Even within these cases, different results can be obtained by altering requirements and tweaking importance. We recognize that actual cases may be more complex, but the process remains the same.

Table 6: Cases Selected For Fixed Wing Design Study

Case	Description
Reconnaissance	Long endurance surveillance – requires some minimum payload, want maximum flight time
Hot Payload Delivery	Deliver a payload as fast as possible – requires minimum payload and endurance, shortest manufacturing time, maximum flight speed
Follow Target	Requires minimum payload, max speed, flight time and acceleration

As an example, a sample set of requirements and importance is presented in Table 7 for the hot payload delivery case.

Table 7: Requirement Priorities for Hot Payload Delivery

	Minimum Payload (kg)	Minimum Endurance (min)	Maximum Manufacturing Time (days)	Minimum Top Speed (m/s)	Minimum Acceleration (m^2/s)
Requirements:	0.1	7	5	5	0.1
Importance (1-100):	1	1	100	100	1

TOPSIS was performed on the filtered solutions to rank order them based on user-specified importance.

RESULTS

Below the following table demonstrate the variety of our solutions, represented by minimum and maximum value or the number of discrete solutions.

Table 8: Ranges for Fixed Wing Input Variables

Variable	Airfoil	Span (mm)	Chord (mm)	Motor	Battery	Elevator half-span (mm)	Elevator chord (mm)	Rudder height (mm)	Rudder chord (mm)	Propeller
Minimum		974.5	151.6			131.0	87.3	87.3	87.3	
Maximum		1149.0	197.6			159.8	106.5	106.5	106.5	
# of Solutions	3	Cont.	Cont.	4	4	Cont.	Cont.	Cont.	Cont.	2

Table 9: Ranges for Fixed Wing Outputs

Variable	Minimum Endurance (min)	Max Payload (kg)	Manufacturing Time (days)	Top Speed (m/s)	Max RoC (m/s)	N ('g')	Max Acceleration (m/s^2)	Launch Speed (m/s)
Minimum	5.26	.1189	3.23	11.38	.971	1.18	.58	6.89
Maximum	10.44	.3467	7.53	24.62	4.996	1.94	4.35	7.98

# of Solutions	Cont.	Cont.	Cont.	Cont.	Cont.	Cont.	Cont.	Cont.

USE CASE RESULTS

In Table 10, the results of the example cases can be compared.

Table 10: Use Case Results

Case	Airfoil	Span (mm)	Battery	Mfg. (days)	Endurance (min)	Motor	Prop	RoC (m/s)	Top Speed (m/s)	Launch Speed (m/s)
Reconnaissance – 300g payload, max flight time	N0012	1113.5	3s5000	6.67	5.6	1250kV	9x7.5	5	23.2	7.87
Reconnaissance – 100g payload, max flight time	S1223	1149	3s5000	4.23	10.3	1000kV	9x7.5	3.6	16.25	7.98
Hot Payload Delivery – 100g payload, max speed, min mfg.	S1223	974.5	3s2200	3.23	5.6	1000kV	11x8	4.1	21.01	7.59
Follow Target - 100g payload, max speed, flight time, acceleration	S1123	1113.5	3s5000	7.29	10.3	1000kV	9x7.5	2.9	20.74	7.78
Follow Target - 100g payload, max speed, acceleration	S1223	1102.4	3s5000	6.25	5.5	1800kV	9x7.5	.97	21.93	7.56

Comparing the first two cases, it can be observed that constraints are activating which provides validation of expectations. The inverse relation between flight time and payload, as well as the inability of our plane with a high lift airfoil to carry high payloads with the possible propeller and motor combinations can also be observed. As expected, the high lift airfoil has worse accelerations, RoC, and speeds, but boasts better endurance. Manufacturing time opposes increases in most of the variables, and it is the limiting factor of span and chord. When prioritizing accelerations, climb speeds, and top speeds, it can be seen that endurance drops and manufacturing time increases, along with a tendency towards higher kV motors.

This examples shown highlight a few of the parameters that are involved, and already a huge amount of variation can be observed. This demonstrates how much freedom the user has to tune the design to their specific needs. With the exact same architecture, a have a huge variety of flight speeds and endurances are available. It is also apparent that adding additional propellers would increase the resolution of the design space, were the necessary data to be made available.

DETAILED DESIGN MODELING

Creo Parametric was selected for our CAD modeling environment for the fixed wing aircraft. Our goal was to drive dimensions and configurations using the final result from TOPSIS. Our driven dimensions and parts include

- Rudder height and chord
- Elevator chord and span
- Battery dimensions (X, Y, Z) – drives battery and battery cage
- Wing span, chord, and airfoil type

In the fixed wing aircraft design, the skeleton was managed from the Excel interface to CREO, but the process for implementing rules driving the parts was similar. For this design the team ignored motor and propeller updates within the CAD software, since those have no impact on the fabricated parts of the plane, and are all visually very similar. Details about the implementation of the CREO based implementation can be found in Appendix A.

FIXED WING AIRCRAFT STUDY CONCLUSIONS

Figures 57 and 58 show images of the final fixed wing aircraft designs produced. The development of this second vehicle using a different toolset allowed verification of the process, and also highlighted the fact that the process is repeatable.

Figure 57: Fixed Wing Aircraft Design

Figure 58: Fixed Wing Aircraft Design

During this second family design, an alternative toolset was able to successfully create a parametric family of vehicles. However, it should be noted that the ARL-Georgia Tech found CREO significantly more difficult to create and manage the product family. This could be to the team's lack of knowledge of the second CAD package, or could indicate that the speed at which the process is implemented may be driven by the toolset. In both cases, however, the final result obtained by creating a modular family of designs easily outweighed the additional engineering required in setting up a modular architecture.

CONCLUSIONS

Over the course of this project the team has developed a toolset that is capable of providing the soldier with a process and toolset capable of allowing an on-demand Micro-Autonomous System service for the soldier. The toolset and associated process fulfils the requirements set out at the beginning of the project and each of the requirements for the individual stakeholders by providing the following:

- Soldier requirements
  - A simplified user interface for vehicle needs to be entered
  - Automated engineering analysis of potential vehicle designs
  - Methods for ranking feasible designs
  - An interface which can be used to select from a prioritized list of feasible designs
  - The use of rapid prototyping tools and automated manufacturing equipment for rapid production
  - An ability to receive a tailored Micro-Autonomous System within 72 hours
- Manufacturing Technician Requirements
  - A simplified interface for entering machine bed sizes and material capabilities
  - A simplified interface for entering parts availabilities
  - A method for updating the component libraries should similar alternative parts become available
  - A automated method for the designs to account for these constraints

By meeting these constraints simultaneously the research team was able to create a process for providing the soldier on-demand Micro-Autonomous Systems. Through additional research, this process could be expanded to other types of systems, and was enabled by the ADAPt Design Process

The ADAPt Design process enabled the MASR vision by creating an automated engineering process for developing derivative designs within a multi-platform product family. This work was tested on a second architecture and the ADAPt design process has applicability beyond the MASR research. Not only does it allow for rapid production of designs, but it could also be expanded to create highly modular components on other systems. For example, a modular mount on a larger vehicle could be created to allow for a multi-product family of attachments. The same process could be modified to produce highly modular larger scale systems as well. As a result, the ADAPt design process is considered one of the key research outcomes of this project.

Through the ADAPt design process, and he interfaces created, the Georgia Tech – ARL collaboration successfully met the objectives of Phase II of the MASR project, and will continue the work within Phase III of the MASR research project.

APPENDIX A – DRIVING AND DRIVEN PARAMETERS FOR FIXED WING AIRCRAFT DESIGN

PASSING DIMENSIONS THROUGH THE ASSEMBLY

Passing dimensions through a top level assembly in Creo is at first confusing, but with a bit of organization becomes trivial. We will first describe the terminology necessary before examining the process in detail.

Creo uses a "Family Table" (similar to a SolidWorks design table), "Parameters," and "Relations" to parameterize the model. Parameters and relations can be altered in something called Pro/Program (Model Intent/Program/Edit Design), where the assembly can be viewed in a code-like environment and additional logic can be implemented.

First, a parameter (with suffix "A") is created in the assembly for every input value listed above. These parameters are then inserted as a separate column into the Family Table, where the values are entered from Excel. To pass parameters down to individual parts so that dimensions can be altered, a separate parameter is created with the same name but suffix "P" in the part file. Then, we assign whatever dimensions we wish to alter a name in the part file, and use relations to equate that dimension to the dimension "P". In the assembly, we use relations to equate the part parameter with the assembly parameter.

As an example, it is useful to look at battery height. In the assembly, one parameter is "BATTHEIGHTA," and accepts a value input to the family table. The user can then open relations and type "BATTHEIGHTP:4 = BATTHEIGHTA." The :4 is the part identification number, is unique to that assembly and is shared between ALL identically named part instances in the assembly. This is very important, as it impacts how we deal with the wing section ends. At the part level, we have "BATTHEIGHT= BATTHEIGHTP," where BATTHEIGHT is our driven dimension. This is repeated for every driven dimension in every driven part. Note that we can drive as many dimensions as we want at the part level with one assembly parameter – we have only 10 inputs from Excel, but drive over 20 different dimensions and part selections. All of our driven dimensions can be seen in Appendix A.

WING MODEL

The wing was by far the most complicated part to model. Because the print envelope is limited to 5.5"x7.5", the wing had to be broken up into sections, and then assembled using two carbon fiber rods. As span changes, the number of wing sections and the length of the end wing sections changes. On top of that, it was necessary to have a model for each airfoil used.

Since identical parts were assigned a parameter, we have individual parts modeled for the center, interior, and end wing sections. We also have separate models for different numbers of wing sections and different airfoils. We modeled 7 and 9 sections for both the N0012 and S1223 airfoils, giving us 4 different subassemblies, with 3 unique wing section parts in each. We use the Pro/Program interface to implement simple IF logic based on the span of the wing and an airfoil string variable. In Creo, it would be encoded as follows.

```
IF WINGSPAN/5.5 < 7 && AIRFOIL == "s1223.dat'"
    ADD SUBASSEMBLY WING_ASMB_7P
    INTERNAL COMPONENT ID 136
    PARENTS = 41(#5) 116(#12)
    END ADD
END IF
```

If the conditions are not fulfilled, the part is never added to the model. Similar techniques can be used for features in parts, but was not done in this particular model.

DRIVING PARAMETERS

DRIVING PARAMETERS

Part name	**Name**	**Description**	**Driving/driven/ configuration**	**Not all dimensions will be driven or driving (e.g. holes for fasteners)** **There is no need to record those here.**
Horizontal tail section	Halfspan		Driving	
	Chord		Driving	
Wing	Airfoil	Selection of wing component geometry	Configuration	
	Span	Span of full wing	Driving	Divide by maximum length of MWS (middle wing section) 5.5", round down to get number of MWS
	Chord		Driving	
Vertical tail section	Height		Driving	
	Chord		Driving	

Battery dimensions	Length		Driving	
	Width		Driving	
	Height		Driving	

DRIVEN PARAMETERS

Dimension name	**Dimension description**	**Driven/ configuration**	**Relation to Driving**
Elevator			
ELEVCHORD	Root chord, from leading edge to trailing edge at the widest point	Driven	Chord
ELEVSPAN	Height, from cutout to tip	Driven	Height
Rudder			
Rudderchord	Root chord, from leading edge to trailing edge at the widest point	Driven	Chord
Rudderheight	Height, from cutout to tip	Driven	Height
End Wing Section			
WINGEND_LENGTH	Total span minus the span of the middle sections, divided by two pieces	Driven	(WINGSPAN-5.5*(#IWS+1))/2
EWS part	Selection of MWS part	Configuration	Wing Airfoil
Chord	Length of wing section (flight direction)	Dimension	Chord
RodHole	Distance from front to rod hole	Dimension	Chord
Interior Wing Section			
IWS part	Selection of MWS part	Configuration	Wing Airfoil
Chord	Length of wing section (flight direction)	Dimension	Chord
Center Wing Section			
Chord	Length of wing section (flight direction)	Dimension	Chord
CWS part	Selection of CWS configuration	Configuration	Wing Airfoil

Wing Assembly			
M132WING_ASMB_7P	Selection of 7 piece wing	Configuration	WINGSPAN
M136WING_ASMB_9P	Selection of 9 piece wing	Configuration	WINGSPAN
Wing Rod			
WINGROD_LENGTH	Length of wing rod	Dimension	WINGSPAN
Wing Bar			
WINGBAR_LENGTH	Length of wing bar	Dimension	WINGSPAN
Battery Sled Protected			
BATTLENGTH	Length of larger cage section	Dimension	Battery length
BATTHEIGHT	Column support length	Dimension	Battery height
BATTHEIGHTB	Column support length	Dimension	Battery height
BATTHEIGHTC	Column support length	Dimension	Battery height
CAGEWIDTH	Widest portion of cage	Dimension	Battery Width
Battery			
BATTLENGTH		Dimension	Battery length
BATTHEIGHT		Dimension	Battery height
BATTWIDTH		Dimension	Battery width

APPENDIX B – REFERENCES

[1]Simpson, Timothy W., Jianxin, Zahed Siddique, and Katja Hölttä-Otto. "A Review of Recent Literature in Product Family Design and Platform-Based Product Development." ***Advances in Product Family and Product Platform Design Methods & Applications*****. Dordrecht: Springer, 2013.**

[2]Forsberg, K. and Mooz, H., Systems Engineering for Faster, Cheapter, Better. 1998

[3]Ananda, c. b. (2015, February 3). UIUC Propeller Database. Retrieved from UIUC Applied Aerodynamics group: http://m-selig.ae.illinois.edu/props/propDB.html